

COMPUTER SCIENCE AND ENGINEERING
B. Tech Computer Science and Engineering

SEMESTER V

Sl. No	Course Code	Course	L-T-P	Category	Page No
1	CST 301	FORMAL LANGUAGES AND AUTOMATA THEORY	3-1-0	PCC	4
2	CST 303	COMPUTER NETWORKS	3-1-0	PCC	16
3	CST 305	SYSTEM SOFTWARE	3-1-0	PCC	28
4	CST 307	MICROPROCESSORS AND MICROCONTROLLERS	3-1-0	PCC	39
5	CST 309	MANAGEMENT OF SOFTWARE SYSTEMS	3-0-0	PCC	49
7	CSL 331	SYSTEM SOFTWARE AND MICROPROCESSORS LAB	0-0-4	PCC	61
8	CSL 333	DATABASE MANAGEMENT SYSTEMS LAB	0-0-4	PCC	66
9	CST 381	CONCEPTS IN SOFTWARE ENGINEERING	3-1-0	Minor	74
10	CST 383	CONCEPTS IN MACHINE LEARNING	3-1-0	Minor	85
11	CST 385	CLIENT SERVER SYSTEMS	3-1-0	Minor	99
12	CST 393	CRYPTOGRAPHIC ALGORITHMS	3-1-0	Honours	110
13	CST 395	NEURAL NETWORKS AND DEEP LEARNING	3-1-0	Honours	122
14	CST 397	PRINCIPLES OF MODEL CHECKING	3-1-0	Honours	135

Sl. No	Course Code	Course	L-T-P	Category	Page No
1	CST 302	COMPILER DESIGN	3-1-0	PCC	147
2	CST 304	COMPUTER GRAPHICS AND IMAGE PROCESSING	3-1-0	PCC	158
3	CST 306	ALGORITHM ANALYSIS AND DESIGN	3-1-0	PCC	170
4	CST 308	COMPREHENSIVE COURSE WORK	1-0-0	PCC	184
5	CSL 332	NETWORKING LAB	0-0-3	PCC	197
6	CSD 334	MINI PROJECT	0-0-3	PCC	203
7	CST 312	FOUNDATIONS OF MACHINE LEARNING	2-1-0	PEC	209
8	CST 322	DATA ANALYTICS	2-1-0	PEC	225
9	CST 332	FOUNDATIONS OF SECURITY IN COMPUTING	2-1-0	PEC	238
10	CST 342	AUTOMATED VERIFICATION	2-1-0	PEC	248
11	CSL 362	PROGRAMMING IN PYTHON	2-1-0	PEC	258
12	CST 372	DATA AND COMPUTER COMMUNICATION	2-1-0	PEC	270
13	CST 382	INTRODUCTION TO SOFTWARE TESTING	3-1-0	Minor	184
14	CST 384	CONCEPTS IN DEEP LEARNING	3-1-0	Minor	295
15	CST 386	WIRELESS NETWORKS AND IOT APPLICATIONS	3-1-0	Minor	309
16	CST 394	NETWORK SECURITY	3-1-0	Honours	321
17	CST 396	ADVANCED TOPICS IN MACHINE LEARNING	3-1-0	Honours	333
18	CST 398	THEORY OF COMPUTABILITY AND COMPLEXITY	3-1-0	Honours	349

APJ ABDUL KALAM
TECHNOLOGICAL
UNIVERSITY

SEMESTER V

KTU



CST 301	FORMAL LANGUAGES AND AUTOMATA THEORY	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble: This is a core course in theoretical computer science. It covers automata and grammar representations for languages in Chomsky Hierarchy. For regular languages, it also covers representations using regular expression and Myhill-Nerode Relation. The topics covered in this course have applications in various domains including compiler design, decidability and complexity theory, software testing, formal modelling and verification of hardware and software.

Prerequisite: Basic knowledge about the following topic is assumed: sets, relations - equivalence relations, functions, proof by Principle of Mathematical Induction.

Course Outcomes: After the completion of the course the student will be able to

CO1	Classify a given formal language into Regular, Context-Free, Context Sensitive, Recursive or Recursively Enumerable. [Cognitive knowledge level: Understand]
CO2	Explain a formal representation of a given regular language as a finite state automaton, regular grammar, regular expression and Myhill-Nerode relation. [Cognitive knowledge level: Understand]
CO3	Design a Pushdown Automaton and a Context-Free Grammar for a given context-free language. [Cognitive knowledge level : Apply]
CO4	Design Turing machines as language acceptors or transducers. [Cognitive knowledge level: Apply]
CO5	Explain the notion of decidability. [Cognitive knowledge level: Understand]

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												

CO2												
CO3												
CO4												
CO5												

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Marks)	Test 2 (Marks)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:Attendance : **10 marks**Continuous Assessment - Test : **25 marks**Continuous Assessment - Assignment : **15 marks****Internal Examination Pattern:**

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus**CST 301 Formal Languages and Automata Theory****Module - 1 (Introduction to Formal Language Theory and Regular Languages)**

Introduction to formal language theory– Alphabets, Strings, Concatenation of strings, Languages.

Regular Languages - Deterministic Finite State Automata (DFA) (Proof of correctness of construction not required), Nondeterministic Finite State Automata (NFA), Equivalence of DFA and NFA, Regular Grammar (RG), Equivalence of RGs and DFA.

Module - 2 (More on Regular Languages)

Regular Expression (RE), Equivalence of REs and DFA, Homomorphisms, Necessary conditions for regular languages, Closure Properties of Regular Languages, DFA state minimization (No proof required).

Module - 3 (Myhill-Nerode Relations and Context Free Grammars)

Myhill-Nerode Relations (MNR)- MNR for regular languages, Myhill-Nerode Theorem (MNT) (No proof required), Applications of MNT.

Context Free Grammar (CFG)- CFG representation of Context Free Languages (proof of correctness is required), derivation trees and ambiguity, Normal forms for CFGs.

Module - 4 (More on Context-Free Languages)

Nondeterministic Pushdown Automata (PDA), Deterministic Pushdown Automata (DPDA), Equivalence of PDAs and CFGs (Proof not required), Pumping Lemma for Context-Free Languages (Proof not required), Closure Properties of Context Free Languages.

Module - 5 (Context Sensitive Languages, Turing Machines)

Context Sensitive Languages - Context Sensitive Grammar (CSG), Linear Bounded Automata.

Turing Machines - Standard Turing Machine, Robustness of Turing Machine, Universal Turing Machine, Halting Problem, Recursive and Recursively Enumerable Languages.

Chomsky classification of formal languages.

Text Book

1. Dexter C. Kozen, Automata and Computability, Springer (1999)

Reference Materials

1. John E Hopcroft, Rajeev Motwani and Jeffrey D Ullman, Introduction to Automata Theory, Languages, and Computation, 3/e, Pearson Education, 2007
2. Michael Sipser, Introduction To Theory of Computation, Cengage Publishers, 2013.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): Identify the class of the following languages in Chomsky Hierarchy:

- $L_1 = \{a^p \mid p \text{ is a prime number}\}$
- $L_2 =$

$\{x \in \{0,1\}^* \mid x \text{ is the binary representation of a decimal number which is a multiple of } 5\}$

- $L_3 = \{a^n b^n c^n \mid n \geq 0\}$
- $L_4 = \{a^m b^n c^{m+n} \mid m > 0, n \geq 0\}$
- $L_5 = \{M \# x \mid M \text{ halts on } x\}$. Here, M is a binary encoding of a Turing Machine and x is a binary input to the Turing Machine.

Course Outcome 2 (CO2):

- (i) Design a DFA for the language $L = \{axb \mid x \in \{a, b\}^*\}$
- (ii) Write a Regular Expression for the language: $L = \{x \in \{a, b\}^* \mid \text{third last symbol in } x \text{ is } b\}$
- (iii) Write a Regular Grammar for the language: $L = \{x \in \{0,1\}^* \mid \text{there are no consecutive zeros in } x\}$
- (iv) Show the equivalence classes of the canonical Myhill-Nerode relation induced by the language: $L = \{x \in \{a, b\}^* \mid x \text{ contains even number of } a\text{'s and odd number of } b\text{'s}\}$.

Course Outcome 3 (CO3):

- (i) Design a PDA for the language $L = \{ww^R \mid w \in \{a, b\}^*\}$. Here, the notation w^R represents the reverse of the string w .
- (ii) Write a Context-Free Grammar for the language $L = \{a^n b^{2n} \mid n \geq 0\}$.

Course Outcome 4 (CO4):

- (i) Design a Turing Machine for the language $L = \{a^n b^n c^n \mid n \geq 0\}$
- (ii) Design a Turing Machine to compute the square of a natural number. Assume that the input is provided in unary representation.

Course Outcome 5 (CO5): Argue that it is undecidable to check whether a Turing Machine M enters a given state during the computation of a given input x .

Model Question paper

QP CODE:

PAGES:3

Reg No: _____

Name : _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST301

Course Name: Formal Languages and Automata Theory

Max.Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Design a DFA for the language $L = \{x \in \{a,b\}^* | aba \text{ is not a substring in } x\}$.
2. Write a Regular Grammar for the language: $L = \{axb | x \in \{a,b\}^*\}$
3. Write a Regular Expression for the language:
 $L = \{x \in \{0,1\}^* | \text{there are no consecutive 1's in } x\}$
4. Prove that the language $L_1 = \{a^{n!} | n \in N\}$ is not regular.
5. List out the applications of Myhill-Nerode Theorem.
6. Write a Context-Free Grammar for the language: $L = \{x \in \{a,b\}^* | \#_a(x) = \#_b(x)\}$. Here, the notation $\#_1(w)$ represents the number of occurrences of the symbol 1 in the string w .
7. Design a PDA for the language of odd length binary palindromes (no explanation is required, just list the transitions in the PDA).
8. Prove that Context Free Languages are closed under set union.
9. Write a Context Sensitive Grammar for the language $L = \{a^n b^n c^n | n \geq 0\}$ (no explanation is required, just write the set of productions in the grammar).

10. Differentiate between Recursive and Recursively Enumerable Languages.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Draw the state-transition diagram showing an NFA N for the following language L . Obtain the DFAD equivalent to N by applying the subset construction algorithm. (7)

$$L = \{x \in \{a, b\}^* | \text{the second last symbol in } x \text{ is } b\}$$

- (b) Draw the state-transition diagram showing a DFA for recognizing the following language: (7)

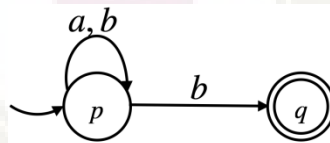
$$L = \{x \in \{0,1\}^* | x \text{ is a binary representation of a natural number which is a multiple of } 5\}$$

OR

12. (a) Write a Regular grammar G for the following language L defined as: $L = \{x \in \{a, b\}^* | x \text{ does not contain consecutive } b\text{'s}\}$. (7)

- (b) Obtain the DFA A_G over the alphabet set $\Sigma = \{a, b\}$, equivalent to the regular grammar G with start symbol S and productions: $S \rightarrow aA$ and $A \rightarrow aA | bA | b$. (7)

13. (a) Using Kleen's construction, obtain the regular expression for the language represented by the following NFA

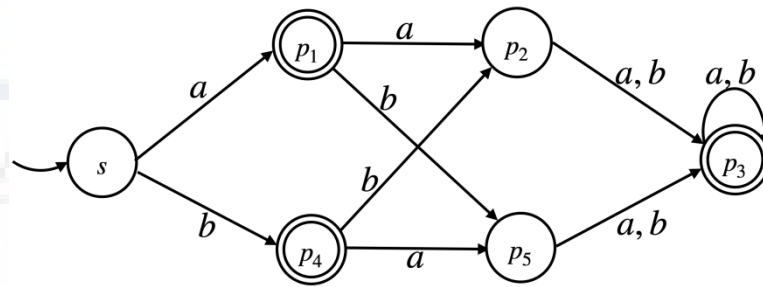


(8)

- (b) Using pumping lemma for regular languages, prove that the language $L = \{a^n b^n | n \geq 0\}$ is not regular. (7)

OR

14. (a)



Obtain the minimum-state DFA from the following DFA. (8)

(b) Using ultimate periodicity for regular languages, prove that the language $L = \{a^{n^2} | n \geq 0\}$ is not regular. (6)

15. (a) Show the equivalence classes of the canonical Myhill-Nerode relation for the language of binary strings with odd number of 1's and even number of 0s. (7)

(b) With an example, explain ambiguity in Context Free Grammar (7)

OR

16. (a) Convert the Context-Free Grammar with productions: $\{S \rightarrow aSb | \epsilon\}$ into Greibach Normal form. (8)

(b) Convert the Context-Free Grammar with productions: $\{S \rightarrow aSa | bSb | SS | \epsilon\}$ into Chomsky Normal form. (6)

17. (a) Design a PDA for the language $L = \{a^m b^n c^{m+n} | n \geq 0, m \geq 0\}$. Also illustrate the computation of the PDA on a string in the language (7)

(b) With an example illustrate how a multi-state PDA can be transformed into an equivalent single-state PDA. (7)

OR

18. (a) Using pumping lemma for context-free languages, prove that the language: $L = \{ww|w \in \{a, b\}^*\}$ is not a context-free language. (6)

(b) With an example illustrate how a CFG can be converted to a single-state PDA (8)

19. (a) Design a Turing machine to obtain the sum of two natural numbers a and b , both represented in unary on the alphabet set $\{1\}$. Assume that initially the tape contains $\vdash 1^a 0 1^b \omega$. The Turing Machine should halt with $\vdash 1^{a+b} \omega$ as the tape content. Also, illustrate the computation of your Turing Machine on the input $a = 3$ and $b = 2$. (7)

(b) With an example illustrate how a CFG can be converted to a single-state PDA. (7)

OR

20. (a) Design a Turing machine to obtain the sum of two natural numbers a and b , both represented in unary on the alphabet set $\{1\}$. Assume that initially the tape contains $\vdash 1^a 0 1^b \omega$. The Turing Machine should halt with $\vdash 1^{a+b} \omega$ as the tape content. Also, illustrate the computation of your Turing Machine on the input $a = 3$ and $b = 2$. (7)

(b) Write a context sensitive grammar for the language $L = \{a^n b^n c^n | n \geq 0\}$. Also illustrate how the the string $a^2 b^2 c^2$ can be derived from the start symbol of the proposed grammar. (7)

Teaching Plan

Sl. No	Topic	No. of Hours (45 hrs)
Module - 1 (Introduction to Formal Language Theory and Regular Languages)		9 Hours
1.1	Introduction to formal language theory – Alphabets, strings, concatenation of strings, Languages	1 Hour
1.2	Deterministic Finite State Automata (DFA) – Example DFA (Proof of correctness of construction not required)	1 Hour
1.3	Formal definition of DFA, Language accepted by the class of DFA	1 Hour
1.4	Nondeterministic Finite State Automata (NFA) – Example NFA	1 Hour
1.5	Formal definition of NFA, NFA with ϵ transitions - examples, formal definition	1 Hour
1.6	Equivalence of DFA and NFA with and without ϵ transitions - Subset construction	1 Hour
1.7	Regular Grammar (RG) – Example RGs, derivation of sentences	1 Hour
1.8	Formal definition of RG, Language represented by a RG	1 Hour
1.9	Equivalence of RG and DFA	1 Hour
Module - 2 (More on Regular Languages)		9 Hours
2.1	Regular Expression (RE) - Example REs and formal definition	1 Hour
2.2	Conversion of RE to NFA with ϵ transition	1 Hour
2.3	Conversion of NFA with ϵ transition to RE (Kleen's construction)	1 Hour
2.4	Homomorphisms	1 Hour
2.5	Pumping Lemma for regular languages	1 Hour
2.6	Ultimate periodicity	1 Hour
2.7	Closure Properties of Regular Languages (proof not required)	1 Hour

2.8	DFA state minimization - Quotient construction	1 Hour
2.9	State Minimization Algorithm - Example	1 Hour
Module - 3 (Myhill-Nerode Relations and Context Free Grammars)		10 Hours
3.1	Myhill-Nerode Relations (MNR) - Example, Properties of MyhillNerode Relation	1 Hour
3.2	Conversion of DFA to MNR (Proof of correctness not required)	1 Hour
3.3	Conversion of MNR to DFA(Proof of correctness not required)	1 Hour
3.4	Myhill-Nerode Theorem (MNT)	1 Hour
3.5	Applications of MNT	1 Hour
3.6	Context Free Grammar (CFG) - Example CFGs and formal definition	1 Hour
3.7	Proving correctness of CFGs	1 Hour
3.8	Derivation Trees and ambiguity	1 Hour
3.9	Chomsky Normal Form	1 Hour
3.10	Greibach Normal Form	1 Hour
Module - 4 (More on Context-Free Languages)		8 Hours
4.1	Nondeterministic Pushdown Automata (PDA) – Example PDAs, formal definition	1 Hour
4.2	Acceptance criteria - equivalence	1 Hour
4.3	Deterministic PDA	1 Hour
4.4	Conversion of CFG to PDA (No proof required)	1 Hour
4.5	Conversion of PDA to CGF - Part I (No proof required)	1 Hour
4.6	Conversion of PDA to CGF - Part II (No proof required)	1 Hour
4.7	Pumping Lemma for context-free languages (No proof required)	1 Hour
4.8	Closure Properties of Context Free Languages	1 Hour

Module - 5 (Context Sensitive Languages, Turing Machines)		9 Hours
5.1	Context Sensitive Grammar (CSG) - Examples, formal definition	1 Hour
5.2	Linear Bounded Automata (LBA) - Example LBA, formal definition	1 Hour
5.3	Turing Machine (TM) - TM as language acceptors - examples, formal definition	1 Hour
5.4	TM as transducers - examples	1 Hour
5.5	Robustness of the standard TM model - Multi-tape TMs, Nondeterministic TM	1 Hour
5.6	Universal Turing Machine	1 Hour
5.7	Halting Problem of TM - proof of its undecidability	1 Hour
5.8	Recursive and Recursively Enumerable Languages	1 Hour
5.9	Chomsky classification of formal languages	1 Hour

CST 303	COMPUTER NETWORKS	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble: Study of this course provides the learners a clear understanding of how computer networks from local area networks to the massive and global Internet are built, how they allow computers to share information and communicate with one another. This course covers the physical aspects of computer networks, layers of OSI Reference model, and inter-networking. The course helps the learners to compare and analyze the existing network technologies and choose a suitable network design for a given system.

Prerequisite: Nil

Course Outcomes: After the completion of the course, the student will be able to

CO#	Course Outcomes
CO1	Explain the features of computer networks, protocols, and network design models (Cognitive Knowledge: Understand)
CO2	Describe the fundamental characteristics of the physical layer and identify the usage in network communication (Cognitive Knowledge: Apply)
CO3	Explain the design issues of data link layer, link layer protocols, bridges and switches (Cognitive Knowledge: Understand)
CO4	Illustrate wired LAN protocols (IEEE 802.3) and wireless LAN protocols (IEEE 802.11) (Cognitive Knowledge: Understand)
CO5	Select appropriate routing algorithms, congestion control techniques, and Quality of Service requirements for a network (Cognitive Knowledge: Apply)
CO6	Illustrate the functions and protocols of the network layer, transport layer, and application layer in inter-networking (Cognitive Knowledge: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓									✓
CO4	✓	✓	✓									✓
CO5	✓	✓	✓	✓								✓
CO6	✓	✓	✓			✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and teamwork
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom's Category	Test 1 (Marks in percentage)	Test 2 (Marks in percentage)	End Semester Examination (Marks in percentage)
Remember	40	30	30

Understand	50	50	50
Apply	10	20	20
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Test : **25 marks**

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus. The second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer anyone. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction and Physical Layer)

Introduction – Uses of computer networks, Network hardware, Network software. Reference models – The OSI reference model, The TCP/IP reference model, Comparison of OSI and TCP/IP reference models.

Physical Layer – Modes of communication, Physical topologies, Signal encoding, Repeaters and hub, Transmission media overview. Performance indicators – Bandwidth, Throughput, Latency, Queuing time, Bandwidth–Delay product.

Module - 2 (Data Link Layer)

Data link layer - Data link layer design issues, Error detection and correction, Sliding window protocols, High-Level Data Link Control(HDLC)protocol. Medium Access Control (MAC) sublayer –Channel allocation problem, Multiple access protocols, Ethernet, Wireless LANs - 802.11, Bridges & switches - Bridges from 802.x to 802.y, Repeaters, Hubs, Bridges, Switches, Routers and Gateways.

Module - 3 (Network Layer)

Network layer design issues. Routing algorithms - The Optimality Principle, Shortest path routing, Flooding, Distance Vector Routing, Link State Routing, Multicast routing, Routing for mobile hosts. Congestion control algorithms. Quality of Service (QoS) - requirements, Techniques for achieving good QoS.

Module - 4 (Network Layer in the Internet)

IP protocol, IP addresses, Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Bootstrap Protocol (BOOTP), Dynamic Host Configuration Protocol (DHCP). Open Shortest Path First(OSPF) Protocol, Border Gateway Protocol (BGP), Internet multicasting, IPv6, ICMPv6.

Module – 5 (Transport Layer and Application Layer)

Transport service – Services provided to the upper layers, Transport service primitives. User Datagram Protocol (UDP). Transmission Control Protocol (TCP) – Overview of TCP, TCP segment header, Connection establishment &release, Connection management modeling, TCP retransmission policy, TCP congestion control.

Application Layer –File Transfer Protocol (FTP), Domain Name System (DNS), Electronic mail, Multipurpose Internet Mail Extension (MIME), Simple Network Management Protocol

(SNMP), World Wide Web(WWW) – Architectural overview.

Text Books

1. Andrew S. Tanenbaum, Computer Networks, 4/e, PHI (Prentice Hall India).
2. Behrouz A Forouzan, Data Communication and Networking, 4/e, Tata McGraw Hill

Reference Books

1. Larry L Peterson and Bruce S Dave, Computer Networks – A Systems Approach, 5/e, Morgan Kaufmann.
2. Fred Halsall, Computer Networking and the Internet, 5/e.
3. James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach, 6/e.
4. Keshav, An Engineering Approach to Computer Networks, Addison Wesley, 1998.
5. W. Richard Stevens. TCP/IP Illustrated Volume 1, Addison-Wesley, 2005.
6. William Stallings, Computer Networking with Internet Protocols, Prentice-Hall, 2004.
7. Request for Comments (RFC) Pages - IETF -<https://www.ietf.org/rfc.html>

Course Level Assessment Questions

Course Outcome1 (CO1)

1. Compare TCP/IP and OSI reference model.
2. The purpose of physical layer is to transport a raw bit stream from one machine to another. Justify.

Course Outcome2 (CO2)

1. Write the physical and transmission characteristics of Optical Fibre Cable guided transmission media.
2. The distance between the sender and receiver systems is about 200 KM. The speed of transmission is 2GB/s. Find out the propagation time?

Course Outcome3 (CO3)

1. Ethernet frames must be at least 64 bytes long to ensure that the transmitter is still going in the event of a collision at the far end of the cable. Fast Ethernet has the same 64-byte minimum frame size but can get the bits out ten times faster. How is it possible to maintain the same minimum frame size?
2. What do you mean by bit stuffing?

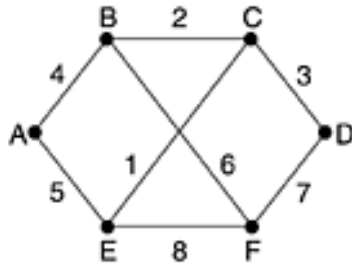
Course Outcome4 (CO4)

1. Draw and explain the frame format for Ethernet.
2. Give the differences between CSMA/CD and CSMA/CA protocol.

Course Outcome5 (CO5)

1. Consider the given subnet in which distance vector routing is used, and the vectors just come in to router C as follows: from B: (5, 0, 8, 12, 6, 2); from D: (16, 12, 6, 0, 9, 10);

and from E: (7, 6, 3, 9, 0, 4). The measured delays from C to B, D, and E, are 6, 3, and 5, respectively. What is C's new routing table? Give both the outgoing line to use and the expected delay.



2. Illustrate the leaky bucket congestion control technique.

Course Outcome 6 (CO6)

1. How do you subnet the Class C IP Address 206.16.2.0 so as to have 30 subnets. What is the subnet mask for the maximum number of hosts? How many hosts can each subnet have?
2. Give the architecture of World Wide Web.

Model Question Paper

QP CODE:

PAGES:

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 303

Course Name : Computer Networks

Max Marks: 100

Duration: 3 Hours

PART-A

(Answer All Questions. Each question carries 3 marks)

1. What does "negotiation" mean when discussing network protocols in a layered architecture? Give an example.

2. Define simplex, half-duplex, and full-duplex transmission modes. Give one example for each.
3. Data link protocols almost always put the CRC in a trailer rather than in a header. Why?
4. An 8-bit byte with binary value 10101111 is to be encoded using an even-parity Hamming code. What is the binary value after encoding?
5. Illustrate the Count to Infinity problem in routing.
6. Describe two major differences between the warning bit method and the Random Early Detection (RED) method.
7. The Protocol field used in the IPv4 header is not present in the fixed IPv6 header. Why?
8. How many octets does the smallest possible IPv6 (IP version 6) datagram contain?
9. Can Transmission Control Protocol(TCP) be used directly over a network (e. g. an Ethernet) without using IP? Justify your answer.
10. When Web pages are sent out, they are prefixed by MIME headers. Why?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) With a neat diagram, explain Open Systems Interconnection (OSI) Reference Model. (8)
 - (b) Compare Twisted Pair, Coaxial Cable and Optical Fibre guided transmission media. (6)
- OR**
12. (a) Consider two networks providing reliable connection-oriented service. One of them offers a reliable byte stream and the other offers a reliable message stream. Are they identical? Justify your answer. (8)
 - (b) Sketch the waveform in Manchester and Differential Manchester Encoding for the bitstream 11000110010. (6)

13. (a) A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is $x^3 + 1$. Show the actual bit string transmitted. Suppose the third bit from the left is inverted during transmission. Show that this error is detected at the receiver's end. (8)

- (b) Explain the working of High-Level Data Link Control (HDLC) protocol. (6)

OR

14. (a) Explain the working of IEEE 802.11 MAC sublayer. (10)

- (b) Distinguish between Bridges and Switches. (4)

15. (a) Illustrate Distance Vector Routing algorithm with an example. (8)

- (b) Explain the characteristics of Routing Information Protocol (RIP). (6)

OR

16. (a) A computer on a 6-Mbps network is regulated by a token bucket. The token bucket is filled at a rate of 1 Mbps. It is initially filled to capacity with 8 megabits. How long can the computer transmit at the full 6 Mbps? (8)

- (b) Explain how routing is performed for mobile hosts. (6)

17. (a) Explain the address resolution problem using Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP) with an example network. (10)

- (b) A network on the Internet has a subnet mask of 255.255.240.0. What is the maximum number of hosts it can handle? (4)

OR

18. (a) How do you subnet the Class C IP address 195.1.1.0 so as to have 10 subnets with a maximum of 12 hosts in each subnet. (6)

- (b) Draw IPv6 Datagram format and explain its features. (8)

19. (a) Distinguish the header formats of Transmission Control protocol (TCP) and User Datagram Protocol (UDP). (8)

- (b) Explain the principal Domain Name System (DNS) resource record types for (6)

IPv4.

OR

20. (a) What is the role of Simple Mail Transfer Protocol (SMTP) in E- mail? (6)
- (b) With the help of a basic model, explain the working of World Wide Web (WWW). (8)

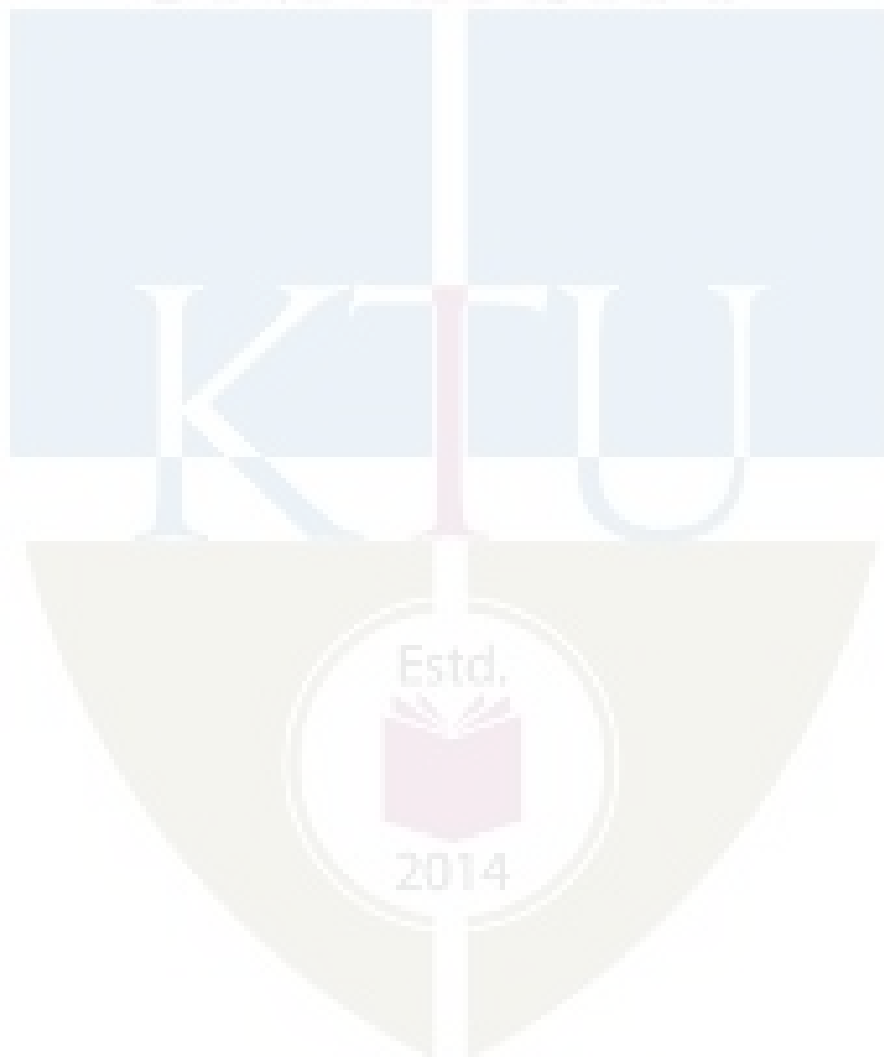
Teaching Plan

No	Contents	No of Lecture Hrs
Module – 1 (Introduction and Physical Layer) (10 hrs)		
1.1	Introduction, Uses of computer networks.	1 hour
1.2	Network Hardware, Local Area Networks (LAN), Metropolitan Area Networks (MAN), Wide Area Networks (WAN), Wireless networks, Home networks, Internetworks.	1 hour
1.3	Network Software, Protocol hierarchies, Design issues for the layers.	1 hour
1.4	Connection-oriented and Connectionless services, Service primitives, Relationship of services to protocols.	1 hour
1.5	Reference models, The OSI reference model.	1 hour
1.6	The TCP/IP reference model, Comparison of OSI and TCP/IP reference models.	1 hour
1.7	Physical layer, Modes of communication, Simplex, Half-duplex, and Full-duplex, Physical topologies, Mesh, Star, Bus, Ring, Hybrid.	1 hour
1.8	Signal encoding, Manchester, Differential Manchester.	1 hour
1.9	Transmission media overview, Guided media (twisted pair, coaxial and fiber optic media), Unguided/wireless media (radio, microwave, and infrared).	1 hour
1.10	Performance indicators, Bandwidth (in Hertz and in Bits per Seconds),	1 hour

	Throughput, Latency (Delay), Queuing time, Bandwidth-Delay product.	
Module 2 – (Data Link Layer) (10 hrs)		
2.1	Data link layer design issues.	1 hour
2.2	Error detection and correction, Error correcting codes	1 hour
2.3	Error detecting codes.	1 hour
2.4	Sliding window protocols.	1 hour
2.5	High-Level Data Link Control(HDLC) protocol.	1 hour
2.6	Medium Access Control (MAC) sublayer, Channel allocation problem, Multiple access protocols.	1 hour
2.7	Ethernet, Ethernet cabling, Manchester encoding, Ethernet MAC sublayer protocol, Binary Exponential Backoff algorithm.	1 hour
2.8	Ethernet performance, Switched Ethernet, Fast Ethernet, Gigabit Ethernet, IEEE 802.2: Logical Link Control.	1 hour
2.9	Wireless LANs, 802.11 protocol stack, Physical layer, MAC Sublayer protocol, Frame structure.	1 hour
2.10	Bridges & switches, Bridges from 802.x to 802.y, Repeaters, Hubs, Bridges, Switches, Routers, and Gateways.	1 hour
Module 3 - (Network Layer) (8 hrs)		
3.1	Network layer design issues.	1 hour
3.2	Routing algorithms, The Optimality Principle, Shortest path routing, Flooding.	1 hour
3.3	Distance Vector Routing.	1 hour
3.4	Link State Routing.	1 hour
3.5	Multicast routing, Routing for mobile hosts.	1 hour

3.6	General principles of congestion control, Congestion prevention policies, Congestion control in virtual circuit subnets.	1 hour
3.7	Congestion control algorithms, Congestion control in Datagram subnets, Load shedding, Jitter control.	1 hour
3.8	Quality of Service, Requirements, Techniques for achieving good Quality of Service.	1 hour
Module 4 – (Network Layer in the Internet) (9 hrs)		
4.1	Network layer in the Internet, Internet Protocol (IP).	1 hour
4.2	IP Addresses, Subnets, Classless Inter-Domain Routing (CIDR).	1 hour
4.3	IP Addresses, Network Address Translation (NAT).	1 hour
4.4	Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP).	1 hour
4.5	Bootstrap Protocol (BOOTP), Dynamic Host Configuration Protocol (DHCP).	1 hour
4.6	Open Shortest Path First (OSPF) protocol.	1 hour
4.7	Border Gateway Protocol (BGP).	1 hour
4.8	Internet multicasting.	1 hour
4.9	IPv6, Header format, Extension headers, Internet Control Message Protocol version 6 (ICMPv6).	1 hour
Module 5 - (Transport Layer and Application Layer) (8 hrs)		
5.1	Transport Service, Services provided to the upper layers, Transport service primitives. User Datagram Protocol (UDP).	1 hour
5.2	Transmission Control Protocol (TCP), TCP segment header, Connection establishment & release, Connection management modeling.	1 hour
5.3	TCP retransmission policy, TCP congestion control.	1 hour
5.4	Application layer, File Transfer Protocol (FTP).	1 hour

5.5	Domain Name System (DNS).	1 hour
5.6	Electronic Mail, Multipurpose Internet Mail Extension (MIME).	1 hour
5.7	Simple Network Management Protocol (SNMP).	1 hour
5.8	World Wide Web, Architectural overview.	1 hour



CST 305	SYSTEM SOFTWARE	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble:

The purpose of this course is to create awareness about the low-level codes which are very close to the hardware and about the environment where programs can be developed and executed. This course helps the learner to understand the machine dependent and machine independent system software features and to design/implement system software like assembler, loader, linker, macroprocessor and device drivers. Study of system software develops ability to design interfaces between software applications and computer hardware.

Prerequisite: A sound knowledge in Data Structures, and Computer Organization

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Distinguish softwares into system and application software categories. (Cognitive Knowledge Level: Understand)
CO2	Identify standard and extended architectural features of machines. (Cognitive Knowledge Level: Apply)
CO3	Identify machine dependent features of system software (Cognitive Knowledge Level: Apply)
CO4	Identify machine independent features of system software. (Cognitive Knowledge Level: Understand)
CO5	Design algorithms for system softwares and analyze the effect of data structures. (Cognitive Knowledge Level: Apply)
CO6	Understand the features of device drivers and editing & debugging tools.(Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	☑	☑			☑							☑
CO2	☑	☑	☑									☑
CO3	☑	☑	☑									☑
CO4	☑	☑										☑
CO5	☑	☑	☑	☑								☑
CO6	☑	☑			☑							☑

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks(%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (Average of series Tests 1&2)	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus**Module-1 (Introduction)**

System Software vs Application Software, Different System Software– Assembler, Linker, Loader, Macro Processor, Text Editor, Debugger, Device Driver, Compiler, Interpreter, Operating System (Basic Concepts only). SIC & SIC/XE Architecture, Addressing modes, SIC & SIC/XE Instruction set , Assembler Directives.

Module-2 (Assembly language programming and Assemblers)

SIC/XE Programming, Basic Functions of Assembler, Assembler Output Format – Header, Text and End Records. Assembler Data Structures, Two Pass Assembler Algorithm, Hand Assembly of SIC/XE Programs.

Module-3 (Assembler Features and Design Options)

Machine Dependent Assembler Features-Instruction Format and Addressing Modes, Program Relocation. Machine Independent Assembler Features –Literals, Symbol Defining Statements, Expressions, Program Blocks, Control Sections and Program Linking. Assembler Design Options- One Pass Assembler, Multi Pass Assembler. Implementation Example-MASM Assembler.

Module-4 (Loader and Linker)

Basic Loader Functions - Design of Absolute Loader, Simple Bootstrap Loader. Machine Dependent Loader Features- Relocation, Program Linking, Algorithm and Data Structures of Two Pass Linking Loader. Machine Independent Loader Features -Automatic Library Search, Loader Options. Loader Design Options.

Module-5 (Macro Preprocessor ,Device driver, Text Editor and Debuggers)

Macro Preprocessor - Macro Instruction Definition and Expansion, One pass Macro processor Algorithm and data structures, Machine Independent Macro Processor Features, Macro processor design options. Device drivers - Anatomy of a device driver, Character and block device drivers, General design of device drivers. Text Editors- Overview of Editing, User Interface, Editor

Structure. Debuggers - Debugging Functions and Capabilities, Relationship with other parts of the system, Debugging Methods- By Induction, Deduction and Backtracking.

Text book

1. Leland L. Beck, System Software: An Introduction to Systems Programming, 3/E, Pearson Education Asia

References

1. D.M. Dhamdhare, Systems Programming and Operating Systems, Second Revised Edition, Tata McGraw Hill.
2. John J. Donovan, Systems Programming, Tata McGraw Hill Edition 1991.
3. George Pajari, Writing UNIX Device Drivers, Addison Wesley Publications (Ebook : <http://tocs.ulb.tu-darmstadt.de/197262074.pdf>).
4. Peter Abel, IBM PC Assembly Language and Programming, Third Edition, Prentice Hall of India.
5. Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, Linux Device Drivers, Third Edition, O.Reilly Books
6. M. Beck, H. Bohme, M. Dziadzka, et al., Linux Kernel Internals, Second Edition, Addison Wesley Publications,
7. J Nithyashri, System Software, Second Edition, Tata McGraw Hill.
8. The C Preprocessor http://gcc.gnu.org/onlinedocs/gcc-2.95.3/cpp_1.html -

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List out two system software and two application software.

Course Outcome 2 (CO2):

1. How is upward compatibility between SIC and SIC/XE machines maintained?
2. Write a sequence of instructions for SIC/XE to divide BETA by GAMMA, setting ALPHA to the integer portion of the quotient and DELTA to the remainder. Use register-to-register instructions to make the calculation as efficient as possible.

Course Outcome 3 (CO3):

1. How do control sections and program blocks differ?
2. Can an assembler incorporating program blocks function using the same data structures as that of a normal two pass assembler? Justify your answer

Course Outcome 4 (CO4):

1. What are literals used for? Does the use of literals change the design of an assembler?

Course Outcome 5 (CO5):

1. Design an assembler that can assemble a source program with different control sections.

Course Outcome 6 (CO6):

1. Describe any one commonly used debugging method.

Model Question Paper

QP CODE:

Reg No: _____

Name: _____

PAGES : 3

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH. DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 305

Course Name: System Software

Max.Marks:100

Duration: 3 Hours

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Differentiate between system software and application software.
2. What are assembler directives? List out any five assembler directives in SIC.
3. Explain the different data structures used in the implementation of Assemblers.
4. List out the functions performed by an assembler.

5. What is a Literal? How is a literal handled by an assembler.
6. What are control sections? What is the advantage of using them?
7. Differentiate between linking loader and linkage editor? Which of these is preferable in a program development environment?
8. What is Automatic Library Search?
9. How should a programmer decide whether to use a macro or a subroutine to accomplish a given logical function?
10. Differentiate between character and block device drivers

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Differentiate between compilers and Interpreters. (4)
- (b) Explain the architecture and addressing modes of SIC machine. (10)

OR

12. (a) Explain the addressing modes supported by SIC/ XE machine with suitable illustrations. (8)
- (b) Explain the difference between (6)
 - i) A1 RESW 3 and A1 WORD 3
 - ii) B BYTE C'23' and B BYTE X'23'
 - iii) END and END LABEL
13. (a) Let NUMBERS be an array of 100 words. Write a sequence of SIC/XE instructions to find the maximum of these numbers. (6)
- (b) Perform hand assembly of the above written program using two pass assembler and show the status of various data structures and object program create. (8)

OR

14. (a) Write down and explain the second pass of a two pass assembler algorithm. (8)
- (b) What is a Program Block. What is its advantage? With suitable example, explain how Program Blocks are handled by SIC assembler. (6)
15. (a) What is a Program Block. What is its advantage? With suitable example, explain how Program Blocks are handled by SIC assembler. (7)
- (b) What is a forward reference? With example, illustrate how forward references are handled by a single pass assembler? (7)

OR

16. (a) With suitable examples explain machine dependent assembler features. (8)
- (b) Explain with examples, the need and working of multipass assembler. (6)
17. (a) With the data structures used, state and explain two pass algorithm for a linking loader. (10)
- (b) Explain about bootstrap loader. (4)

OR

18. (a) Explain about machine independent loader features (9)
- (b) What is Dynamic Linking? With example, illustrate how dynamic linking is performed. (5)
19. (a) Write down the single pass macro processor algorithm and with suitable example illustrate its working. (10)
- (b) How are unique labels generated during Macro Expansion? (4)

OR

20. (a) Explain Text Editor structure in detail with a neat diagram. (7)

(b) Explain the different debugging methods in detail.

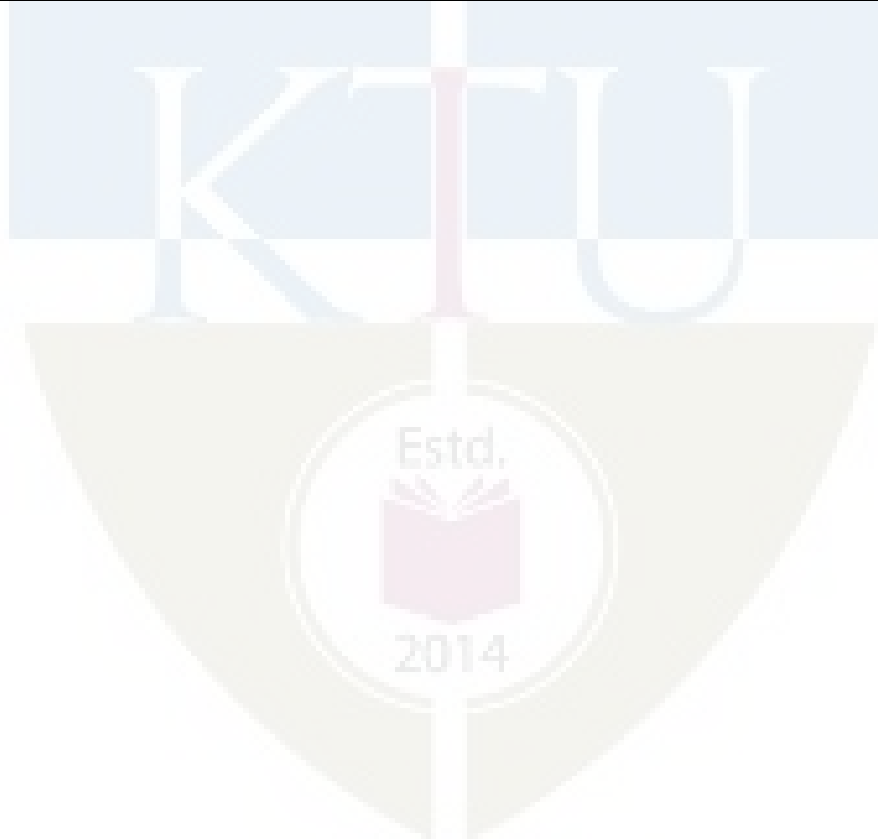
(7)

Teaching Plan

No	Contents	No: of Lecture Hours
Module -1 (Introduction) (9 hours)		
1.1	System Software Vs. Application Software , Different System Software– Assembler, Linker, Loader, Macro Processor	1 hour
1.2	Text Editor, Debugger, Device Driver, Compiler, Interpreter, Operating System(Basic Concepts only)	1 hour
1.3	SIC Architecture	1 hour
1.4	SIC Addressing modes	1 hour
1.5	SIC Instruction set & Assembler directives	1 hour
1.6	SIC/XE Architecture	1 hour
1.7	SIC/XE Instruction format	1 hour
1.8	SIC/XE Addressing modes	1 hour
1.9	SIC/XE Instruction set	1 hour
Module -2 (Assembly language programming and Assemblers) (8 hours)		
2.1	SIC Programming	1 hour
2.2	SIC/XE Programming	1 hour
2.3	Basic Functions of Assembler	1 hour
2.4	Assembler output format – Header, Text and End Records	1 hour
2.5	Assembler data structures	1 hour
2.6	Pass 1 of two pass SIC assembler algorithm	1 hour
2.7	Pass 2 of two pass SIC assembler algorithm	1 hour
2.8	Hand assembly of SIC Program	1 Hour
Module-3 (Assembler design options)(11 hours)		

3.1	Machine dependent assembler features-Instruction format and addressing modes, program relocation	1 hour
3.2	Hand assembly of SIC/XE program	1 Hour
3.3	Machine Independent assembler features – Literals	1 hour
3.4	Machine Independent assembler features – Symbol defining statements, expression	1 hour
3.5	Machine Independent assembler features – program blocks	1 hour
3.6	Machine Independent assembler features – program blocks illustration with examples	1 hour
3.7	Machine Independent assembler features – Control sections and program linking.	1 hour
3.8	Machine Independent assembler features – Control sections and program linking. Illustration with example	1 hour
3.9	Assembler design options- One Pass assembler	1 hour
3.10	Multi pass assembler	1 hour
3.11	Implementation example: MASM Assembler	1 hour
Module-4 (Linker and Loader) (8 hours)		
4.1	Basic Loader functions - Design of absolute loader	1 hour
4.2	Simple bootstrap Loader	1 hour
4.3	Machine dependent loader features- Relocation	1 hour
4.4	Machine dependent loader features- Program Linking algorithm and data structures of First pass of two pass Linking Loader	1 hour
4.5	Machine dependent loader features- Program Linking algorithm and data structures of Second pass of two pass Linking Loader	1 hour
4.6	Machine independent loader feature - Automatic library search	1 hour
4.7	Machine independent loader features - Loader options	1 hour
4.8	Loader Design Option- Linking Loader, Linkage Editor, Dynamic Linking	1 hour
Module –5 (Macro Preprocessor, Device drivers, Text Editors, Debuggers) (9 hours)		
5.1	Macro Preprocessor- Macro Instruction Definition and Expansion	1 hour

5.2	One pass Macro processor algorithm and data structures	1 hour
5.3	One pass Macro processor Algorithm and data structures illustration with example	1 hour
5.4	Machine Independent Macro Processor Features- generation of unique labels, Concatenation of macro parameter, Keyword macro parameters	1 hour
5.5	Machine Independent Macro Processor Features- Conditional Macro Expansion	1 hour
5.6	Macro processor design options	1 hour
5.7	Device drivers- Anatomy of a device driver, Character and block device drivers, General design of device drivers	1 hour
5.8	Text Editors- Overview of Editing, User Interface , Editor Structure	1 hour
5.9	Debuggers :- Debugging Functions and Capabilities, Debugging Methods- By Induction, Deduction and Backtracking.	1 hour



CST 307	MICROPROCESSORS AND MICROCONTROLLERS	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble: The course enables the learners capable of understanding the fundamental architecture of microprocessors and micro controllers. This course focuses on the architecture, assembly language programming, interrupts, interfacing of microprocessors with peripheral devices and microcontrollers and its programming. It helps the learners to extend the study of latest processors and develop hardware based solutions.

Prerequisite : Sound knowledge in Logic System Design and Computer organization & architecture.

CO#	Course Outcomes
CO1	Illustrate the architecture , modes of operation and addressing modes of microprocessors (Cognitive knowledge: Understand)
CO2	Develop 8086 assembly language programs. (Cognitive Knowledge Level: Apply)
CO3	Demonstrate interrupts, its handling and programming in 8086. (Cognitive Knowledge Level: Apply)
CO4	Illustrate how different peripherals (8255,8254,8257) and memory are interfaced with microprocessors. (Cognitive Knowledge Level: Understand)
CO5	Outline features of microcontrollers and develop low level programs. (Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test1 (%)	Test2 (%)	
Remember	20	20	20
Understand	40	40	40
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	:	10 marks
Continuous Assessment Tests	:	25 marks
Continuous Assessment Assignment	:	15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus**Module-1(Evolution of microprocessors):**

8085 microprocessor (-Basic Architecture only). 8086 microprocessor – Architecture and signals, Physical Memory organization, Minimum and maximum mode of 8086 system and timings. Comparison of 8086 and 8088. Machine language Instruction format.

Module-2 (Addressing modes and instructions):

Addressing Modes of 8086. Instruction set – data copy /transfer instructions, arithmetic instructions, logical instructions, string manipulation instructions, branch instructions, unconditional and conditional branch instruction, flag manipulation and processor control instructions. Assembler Directives and operators. Assembly Language Programming with 8086.

Module- 3 (Stack and interrupts):

Stack structure of 8086, programming using stack- Interrupts - Types of Interrupts and Interrupt Service Routine- Handling Interrupts in 8086- Interrupt programming. -

Programmable Interrupt Controller - 8259, Architecture (Just mention the control word, no need to memorize the control word)- Interfacing Memory with 8086.

Module- 4 (Interfacing chips):

Programmable Peripheral Input/output port 8255 - Architecture and modes of operation- Programmable interval timer 8254-Architecture and modes of operation- DMA controller 8257 Architecture (Just mention the control word, no need to memorize the control word of 8254 and 8257)

Module- 5 (Microcontrollers):

8051 Architecture- Register Organization- Memory and I/O addressing- Interrupts and Stack- 8051 Addressing Modes- Instruction Set- data transfer instructions, arithmetic instructions, logical instructions, Boolean instructions, control transfer instructions- Simple programs.

Text Books

1. Bhurchandi and Ray, Advanced Microprocessors and Peripherals, Third Edition McGraw Hill.
2. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, Pearson Education.
3. Ramesh Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, Penram International Publishing Pvt. Ltd.

Reference Books

1. Barry B. Brey, The Intel Microprocessors – Architecture, Programming and Interfacing, Eighth Edition, Pearson Education.
2. A. NagoorKani, Microprocessors and Microcontrollers, Second Edition, Tata McGraw Hill
3. Douglas V. Hall, SSSP Rao, Microprocessors and Interfacing, Third Edition, McGrawHill Education.

Sample Course Level Assessment Questions

Course Outcome1 (CO1):

- 1) Describe how pipelining is implemented in 8086 microprocessor
- 2) Illustrate maximum mode signals in 8086.

Course Outcome 2(CO2):

- 1) Write an 8086 assembly language program for sorting a sequence of N, 8 bit numbers. Describe the modifications that can be done on the above program so that it will sort N, 16 bit numbers. Rewrite the program with those modifications also.

Course Outcome 3 (CO3):

- 1) Design an interface between 8086 CPU and two chips of 16 x 8 EPROM and two chips of 32K x 8 RAM. Select the starting address of EPROM suitably. The RAM address must start at 00000H.
- 2) Give the sequence of instructions for setting the IVT for interrupt type 23H. Assume the Interrupt Service Routine, is present in the code segment named CODE.
- 3) Describe the role of Interrupt Request register and In service register in 8259.

Course Outcome 4(CO4):

- 1) Show how to interface an 8255 with 8086 to work as an I/O port with the following specifications. Initialize port A as output, port B as input and port C as output. Port A address should be 05A0H. Write a program to sense switch positions SW 0 -SW 7 connected to port B. The sensed pattern is to be displayed on port A, to which 8 LED's are attached, while port C lower displays *number of off switches* out of total 8 switches.
- 2) Specify the importance of the DMA address register and Terminal count register in 8257.

Course Outcome 5(CO5):

- 1) Write an 8051 assembly language program to count the number of 1's and 0's in a given 8 bit number
- 2) Write an 8051 assembly language program for computing the square root of an 8 bit number.

Model Question Paper

QP CODE:

Reg No: _____

Name: _____

PAGES : 4

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH. DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 307

Course Name: Microprocessors and Microcontrollers

Max.Marks:100

Duration: 3 Hours

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Describe the functions of following signals in 8086
a)NMI b)ALE
2. List any three differences between 8085 and 8086 microprocessors.
3. Assume AL register is having the value 7FH. What will be the content of AL after the following instructions are executed
a)ROR AL,01 b)SAR AL,01
4. Specify the use of following assembler directives - EQU , EVEN
5. Differentiate between maskable and non maskable interrupts?
6. Define Interrupt Service Routine? How to find the address of the ISR corresponding to a given interrupt in 8086?
7. Give the purposes of the signals DRQ, TC and MARK in 8257?
8. How 8254 is used as a square wave generator?
9. Differentiate between indirect and indexed addressing modes in 8051.

10. Write the sequence of 8051 instructions to store any two numbers at two consecutive locations 70H and 71H, multiply them and store the result in location 72H. (10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Specify the significance of segmentation and how it is implemented in 8086 (5)
 (b) Explain the maximum mode signals in 8086. (9)

OR

12. (a) Write down the differences between 8086 and 8088 processors (4)
 (b) Explain the physical memory organization of 8086 with a neat diagram. How does the 8086 processor access a word from an odd memory location? How many memory cycles does it take? (10)
13. (a) Write an 8086 assembly language program for finding the sum of the squares of first N natural numbers. Calculate the squares of each number using a subroutine SQUARE. (10)
 (b) Describe any four control transfer instructions in 8086. (4)

OR

14. (a) Write an 8086 assembly language program for printing the reverse of a given input string. (5)
 (b) Explain the addressing modes for sequential control flow instructions in 8086. (9)
15. (a) Give the stack structure of 8086. (5)
 (b) Explain the architecture of 8259 with diagram (9)

OR

16. (a) Interface 32Kx8 RAM using four numbers of 8Kx8 memory chips and 16Kx8 ROM using two numbers of 8Kx8 EPROM chips. The address map is given as RAM starts at 00000H and ROM ends at FFFFFH (10)
 (b) Describe the predefined interrupts in 8086 (4)

17. (a) Explain the architecture of 8255 with a neat diagram (10)
- (b) Identify the mode and I/O configuration for ports A, B and C of an 8255 after its control register is loaded with 86 H? (4)

OR

18. (a) Define Direct Memory Access (DMA) and illustrate the role of a DMA controller? Explain the register organization of 8257 and state how these registers are used during DMA transfer operations. (8)
- (b) Explain the architecture of 8254 timer chip (6)
19. (a) Explain the architecture of 8051 microcontroller. (9)
- (b) Write an 8051 assembly language program for adding two matrices whose elements are stored sequentially in some memory location. Assume suitable locations. (5)

OR

20. (a) Explain the internal data memory organization of 8051. (9)
- (b) Describe the control transfer instructions of 8051 microcontroller. (5)

Teaching Plan

No	Contents	No of Lecture Hrs
Module 1 : (Evolution of microprocessors) (9 hours)		
1.1	Overview of 8085 microprocessor	1 hour
1.2	Architecture of 8085	1 hour
1.3	Architecture of 8086	1 hour
1.4	Signals in 8086	1 hour
1.5	Physical Memory organization	1 hour
1.6	Minimum and maximum mode 8086 system and timings(Lecture 1)	1 hour
1.7	Minimum and maximum mode 8086 system and timings(Lecture 2)	1 hour
1.8	Comparison of 8086 and 8088	1 hour
1.9	Machine language Instruction format	1 hour
Module 2 :(programming of 8086) (9 hours)		
2.1	Addressing Modes of 8086	1 hour
2.2	Instruction set – data copy/transfer instructions	1 hour
2.3	arithmetic instructions, logical instructions	1 hour
2.4	string manipulation instructions, branch instructions	1 hour
2.4	unconditional and conditional branch instruction	1 hour
2.5	flag manipulation and processor control instructions	1 hour
2.6	Assembler Directives and operators	1 hour
2.7	Assembly Language Programming with 8086(Lecture 1)	1 hour
2.8	Assembly Language Programming with 8086(Lecture 2)	1 hour
2.9	Assembly Language Programming with 8086(Lecture 3)	1 hour
Module 3 : (stack and Interrupts) (9 hours)		
3.1	Stack structure of 8086, programming using stack.	1 hour
3.2	Types of Interrupts and Interrupt Service Routine.	1 hour
3.3	Handling Interrupts in 8086(Lecture 1)	1 hour
3.4	Handling Interrupts in 8086(Lecture 2)	1 hour

3.5	Interrupt programming.	1 hour
3.6	Programmable Interrupt Controller -8259 (Lecture 1)	1 hour
3.7	Programmable Interrupt Controller -8259 (Lecture 2)	1 hour
3.8	Interfacing Memory with 8086 (Lecture 1)	1 hour
3.9	Interfacing Memory with 8086 (Lecture 2)	1 hour
Module 4 :(Interfacing chips) (7 hours)		
4.1	Programmable Peripheral Input/output port- 8255 (Lecture 1)	1 hour
4.2	Programmable Peripheral Input/output port- 8255 (Lecture 2)	1 hour
4.3	Programmable Peripheral Input/output port- 8255 (Lecture 3)	1 hour
4.4	Programmable interval timer 8254 (Lecture 1)	1 hour
4.5	Programmable interval timer 8254 (Lecture 2)	1 hour
4.6	DMA controller 8257 Architecture (Lecture 1)	1 hour
4.7	DMA controller 8257 Architecture (Lecture 2)	1 hour
Module 5 : (Microcontrollers) (11 hours)		
5.1	8051 Architecture (Lecture 1)	1 hour
5.2	8051 Architecture (Lecture 2)	1 hour
5.3	Register Organization, Memory and I/O addressing	1 hour
5.4	Interrupts and Stack	1 hour
5.5	Addressing Modes	1 hour
5.6	Data transfer instructions, Arithmetic instructions	1 hour
5.7	Logical instructions,	1 hour
5.8	Boolean instructions	1 hour
5.9	Control transfer instructions	1 hour
5.10	Programming of 8051 (Lecture 1)	1 hour
5.11	Programming of 8051(Lecture 2)	1 hour

CST 309	MANAGEMENT OF SOFTWARE SYSTEMS	Category	L	T	P	Credit	Year of Introduction
		PCC	3	0	0	3	2019

Preamble: This course provides fundamental knowledge in the Software Development Process. It covers Software Development, Quality Assurance, Project Management concepts and technology trends. This course enables the learners to apply state of the art industry practices in Software development.

Prerequisite: Basic understanding of Object Oriented Design and Development.

Course Outcomes: After the completion of the course the student will be able to

CO1	Demonstrate Traditional and Agile Software Development approaches (Cognitive Knowledge Level: Apply)
CO2	Prepare Software Requirement Specification and Software Design for a given problem. (Cognitive Knowledge Level: Apply)
CO3	Justify the significance of design patterns and licensing terms in software development, prepare testing, maintenance and DevOps strategies for a project. (Cognitive Knowledge Level: Apply)
CO4	Make use of software project management concepts while planning, estimation, scheduling, tracking and change management of a project, with a traditional/agile framework. (Cognitive Knowledge Level: Apply)
CO5	Utilize SQA practices, Process Improvement techniques and Technology advancements in cloud based software models and containers & microservices. (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓						✓
CO2	✓	✓	✓	✓		✓				✓	✓	✓
CO3	✓	✓	✓	✓				✓		✓	✓	✓
CO4	✓	✓	✓	✓		✓			✓	✓	✓	✓
CO5	✓	✓	✓	✓		✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	40	40	50
Apply	30	30	20
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks (Each student shall identify a software development problem and prepare Requirements Specification, Design Document, Project Plan and Test case documents for the identified problem as the assignment.)

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks.

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 subdivisions and carries 14 marks.

Syllabus

Module 1 : Introduction to Software Engineering (7 hours)

Introduction to Software Engineering - Professional software development, Software engineering ethics. Software process models - The waterfall model, Incremental development. Process activities - Software specification, Software design and implementation, Software validation, Software evolution. Coping with change - Prototyping, Incremental delivery, Boehm's Spiral Model. Agile software development - Agile methods, agile manifesto - values and principles. Agile development techniques, Agile Project Management. Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care.

Module 2 : Requirement Analysis and Design (8 hours)

Functional and non-functional requirements, Requirements engineering processes. Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix. Developing use cases, Software Requirements Specification Template, Personas, Scenarios, User stories, Feature identification. Design concepts - Design within the context of software engineering, Design Process, Design concepts, Design Model. Architectural Design - Software Architecture, Architectural Styles, Architectural considerations, Architectural Design Component level design - What is a component?, Designing Class-Based Components, Conducting Component level design, Component level design for web-apps. Template of a Design Document as per “IEEE Std 1016-2009 IEEE Standard for Information Technology Systems Design Software Design Descriptions”. Case study: The Ariane 5 launcher failure.

Module 3 : Implementation and Testing (9 hours)

Object-oriented design using the UML, Design patterns, Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD. Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. Informal Review, Formal Technical Reviews, Post-mortem evaluations. Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing, Debugging, White box testing, Path testing, Control Structure testing, Black box testing, Testing Documentation and Help facilities. Test automation, Test-driven development, Security testing. Overview of DevOps and Code Management - Code management, DevOps automation, Continuous Integration, Delivery, and Deployment (CI/CD/CD). Software Evolution - Evolution processes, Software maintenance.

Module 4 : Software Project Management (6 hours)

Software Project Management - Risk management, Managing people, Teamwork. Project Planning, Software pricing, Plan-driven development, Project scheduling, Agile planning. Estimation techniques, COCOMO cost modeling. Configuration management, Version management, System building, Change management, Release management, Agile software management - SCRUM framework. Kanban methodology and lean approaches.

Module 5 : Software Quality, Process Improvement and Technology trends (6 hours)

Software Quality, Software Quality Dilemma, Achieving Software Quality Elements of Software Quality Assurance, SQA Tasks, Software measurement and metrics. Software Process Improvement(SPI), SPI Process CMMI process improvement framework, ISO 9001:2000 for Software. Cloud-based Software - Virtualisation and containers, Everything as a service(IaaS, PaaS), Software as a service. Microservices Architecture - Microservices, Microservices architecture, Microservice deployment.

Text Books

1. Book 1 - Ian Sommerville, Software Engineering, Pearson Education, Tenth edition, 2015.
2. Book 2 - Roger S. Pressman, Software Engineering : A practitioner's approach, McGraw Hill publication, Eighth edition, 2014
3. Book 3 - Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson Education, First Edition, 2020.

References

1. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
2. IEEE Std 1016-2009 IEEE Standard for Information Technology—Systems Design—Software Design Descriptions

3. David J. Anderson, Kanban, Blue Hole Press 2010
4. David J. Anderson, Agile Management for Software Engineering, Pearson, 2003
5. Walker Royce, Software Project Management : A unified framework, Pearson Education, 1998
6. Steve. Denning, The age of agile, how smart companies are transforming the way work gets done. New York, Amacom, 2018.
7. Satya Nadella, Hit Refresh: The Quest to Rediscover Microsoft's Soul and Imagine a Better Future for Everyone, Harper Business, 2017
8. Henrico Dolfing, Project Failure Case Studies: Lessons learned from other people's mistakes, Kindle edition
9. Mary Poppendieck, Implementing Lean Software Development: From Concept to Cash, Addison-Wesley Signature Series, 2006
10. StarUML documentation - <https://docs.staruml.io/>
11. OpenProject documentation - <https://docs.openproject.org/>
12. BugZilla documentation - <https://www.bugzilla.org/docs/>
13. GitHub documentation - <https://guides.github.com/>
14. Jira documentation - <https://www.atlassian.com/software/jira>

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are the advantages of an incremental development model over a waterfall model?
2. Illustrate how the process differs in agile software development and traditional software development with a socially relevant case study. (Assignment question)

Course Outcome 2 (CO2):

1. How to prepare a software requirement specification?
2. Differentiate between Architectural design and Component level design.
3. How does agile approaches help software developers to capture and define the user requirements effectively?
4. What is the relevance of the SRS specification in software development?
5. Prepare a use case diagram for a library management system.

Course Outcome 3 (CO3):

1. Differentiate between the different types of software testing strategies.
2. Justify the need for DevOps practices?
3. How do design patterns help software architects communicate the design of a complex system effectively?

4. What are the proactive approaches one can take to optimise efforts in the testing phase?

Course Outcome 4 (CO4):

1. Illustrate the activities involved in software project management for a socially relevant problem?
2. How do SCRUM, Kanban and Lean methodologies help software project management?
3. Is rolling level planning in software project management beneficial? Justify your answer.
4. How would you assess the risks in your software development project? Explain how you can manage identified risks?

Course Outcome 5 (CO5):

1. Justify the importance of Software Process improvement?
2. Explain the benefits of cloud based software development, containers and microservices.
3. Give the role of retrospectives in improving the software development process.
4. Illustrate the use of project history data as a prediction tool to plan future socially relevant projects.

Model Question Paper

QP CODE:

Reg No: _____

Name : _____

PAGES : 3

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FIFTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR**

Course Code: CST 309

Course Name: Management of Software Systems

Duration: 3 Hrs

Max. Marks :100

PART A

Answer all Questions. Each question carries 3 marks

1. Why professional software that is developed for a customer is not simply the programs that have been developed and delivered.
2. Incremental software development could be very effectively used for customers who do not have a clear idea about the systems needed for their operations. Justify.
3. Identify any four types of requirements that may be defined for a software system
4. Describe software architecture
5. Differentiate between GPL and LGPL?
6. Compare white box testing and black box testing.
7. Specify the importance of risk management in software project management?
8. Describe COCOMO cost estimation model.
9. Discuss the software quality dilemma
10. List the levels of the CMMI model? (10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Compare waterfall model and spiral model (8)

- (b) Explain Agile ceremonies and Agile manifesto (6)
12. (a) Illustrate software process activities with an example. (8)
- (b) Explain Agile Development techniques and Agile Project Management (6)
13. (a) What are functional and nonfunctional requirements? Imagine that you are developing a library management software for your college, list eight functional requirements and four nonfunctional requirements. (10)
- (b) List the components of a software requirement specification? (4)
- OR**
14. (a) Explain Personas, Scenarios, User stories and Feature identification? (8)
- (b) Compare Software Architecture design and Component level design (6)
15. (a) Explain software testing strategies. (8)
- (b) Describe the formal and informal review techniques. (6)
- OR**
16. (a) Explain Continuous Integration, Delivery, and Deployment CI/CD/CD) (8)
- (b) Explain test driven development (6)
17. (a) What is a critical path and demonstrate its significance in a project schedule with the help of a sample project schedule. (8)
- (b) Explain plan driven development and project scheduling. (6)
- OR**
18. (a) Explain elements of Software Quality Assurance and SQA Tasks. (6)
- (b) What is algorithmic cost modeling? What problems does it suffer from when (8)

compared with other approaches to cost estimation?

19. (a) Explain elements of Software Quality Assurance and SQA Tasks. (8)

(b) Illustrate SPI process with an example. (6)

OR

20. (a) Compare CMMI and ISO 9001:2000. (8)

(b) How can Software projects benefit from Container deployment and Micro service deployment? (6)

Teaching Plan

No	Contents	No of Lecture Hrs
Module 1 : Introduction to Software Engineering (7 hours)		
1.1	Introduction to Software Engineering.[Book 1, Chapter 1]	1 hour
1.2	Software process models [Book 1 - Chapter 2]	1 hour
1.3	Process activities [Book 1 - Chapter 2]	1 hour
1.4	Coping with change [Book 1 - Chapter 2, Book 2 - Chapter 4]	1 hour
1.5	Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care. [Book 1 - Chapter 1]	1 hour
1.6	Agile software development [Book 1 - Chapter 3]	1 hour
1.7	Agile development techniques, Agile Project Management.[Book 1 - Chapter 3]	1 hour
Module 2 : Requirement Analysis and Design (8 hours)		
2.1	Functional and non-functional requirements, Requirements engineering processes [Book 1 - Chapter 4]	1 hour
2.2	Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix [Book 1 - Chapter 4]	1 hour
2.3	Developing use cases, Software Requirements Specification Template [Book 2 - Chapter 8]	1 hour

2.4	Personas, Scenarios, User stories, Feature identification [Book 3 - Chapter 3]	1 hour
2.5	Design concepts [Book 2 - Chapter 12]	1 hour
2.6	Architectural Design [Book 2 - Chapter 13]	1 hour
2.7	Component level design [Book 2 - Chapter 14]	1 hour
2.8	Design Document Template. Case study: The Ariane 5 launcher failure. [Ref - 2, Book 2 - Chapter 16]	1 hour
Module 3 : Implementation and Testing (9 hours)		
3.1	Object-oriented design using the UML, Design patterns [Book 1 - Chapter 7]	1 hour
3.2	Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD [Book 1 - Chapter 7]	1 hour
3.3	Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. [Book 2 - Chapter 20]	1 hour
3.4	Informal Review, Formal Technical Reviews, Post-mortem evaluations. [Book 2 - Chapter 20]	1 hour
3.5	Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing and Debugging (basic concepts only). [Book 2 - Chapter 22]	1 hour
3.6	White box testing, Path testing, Control Structure testing, Black box testing. Test documentation [Book 2 - Chapter 23]	1 hour
3.7	Test automation, Test-driven development, Security testing. [Book 3 - Chapter 9]	1 hour
3.8	DevOps and Code Management - Code management, DevOps automation, CI/CD/CD. [Book 3 - Chapter 10]	1 hour
3.9	Software Evolution - Evolution processes, Software maintenance. [Book 1 - Chapter 9]	1 hour
Module 4 : Software Project Management (6 hours)		
4.1	Software Project Management - Risk management, Managing people, Teamwork [Book 1 - Chapter 22]	1 hour
4.2	Project Planning - Software pricing, Plan-driven development, Project scheduling, Agile planning [Book 1 - Chapter 23]	1 hour
4.3	Estimation techniques [Book 1 - Chapter 23]	1 hour
4.4	Configuration management [Book 1 - Chapter 25]	1 hour

4.5	Agile software management - SCRUM framework [Book 2 - Chapter 5]	1 hour
4.6	Kanban methodology and lean approaches.[Ref 9 - Chapter 2]	1 hour
Module 5 : Software Quality, Process Improvement and Technology trends (6 hours)		
5.1	Software Quality, Software Quality Dilemma, Achieving Software Quality. [Book 2 - Chapter 19]	1 hour
5.2	Elements of Software Quality Assurance, SQA Tasks , Software measurement and metrics. [Book 3 - Chapter 21]	1 hour
5.3	Software Process Improvement (SPI), SPI Process [Book 2 - Chapter 37]	1 hour
5.4	CMMI process improvement framework, ISO 9001:2000 for Software. [Book 2 - Chapter 37]	1 hour
5.5	Cloud-based Software - Virtualisation and containers, IaaS, PaaS, SaaS.[Book 3 - Chapter 5]	1 hour
5.6	Microservices Architecture - Microservices, Microservices architecture, Microservice deployment [Book 3 - Chapter 6]	1 hour

CSL 331	SYSTEM SOFTWARE AND MICROPROCESSORS LAB	Category	L	T	P	Credit	Year of Introduction
		PCC	0	0	4	2	2019

Preamble: The aim of this course is to give hands-on experience in how microcontrollers, and microprocessors can be programmed. The course also aims to enable students to design and implement system software. The student should get familiar with assembly level programming of microprocessors and microcontrollers, interfacing of devices to microcontrollers, resource allocation algorithms in operating systems and design and implementation of system software.

Prerequisite: Sound knowledge in Operating systems

CO1	Develop 8086 programs and execute it using a microprocessor kit. (Cognitive Knowledge Level: Apply) .
CO2	Develop 8086 programs and, debug and execute it using MASM assemblers (Cognitive Knowledge Level: Apply)
CO3	Develop and execute programs to interface stepper motor, 8255, 8279 and digital to analog converters with 8086 trainer kit (Cognitive Knowledge Level: Apply)
CO4	Implement and execute different scheduling and paging algorithms in OS (Cognitive Knowledge Level: Apply)
CO5	Design and implement assemblers, Loaders and macroprocessors. (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	☑	☑	☑	☑				☑		☑		☑
CO2	☑	☑	☑	☑				☑		☑		☑
CO3	☑	☑	☑	☑				☑		☑		☑
CO4	☑	☑	☑	☑				☑		☑		☑
CO5	☑	☑	☑	☑				☑		☑		☑

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Test (Internal Exam) Percentage	End Semester Examination Percentage
Remember	20	20
Understand	20	20
Apply	60	60
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 15 marks
Continuous Evaluation in Lab	: 30 marks
Continuous Assessment Test	: 15 marks
Viva-voce	: 15 marks

Internal Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks will be converted out of 75 for End Semester Examination.

Operating System to Use in Lab : Linux

Compiler/Software to Use in Lab : gcc

Programming Language to Use in Lab : Ansi C

Any compatible assembler can be used for implementation of 8086 programs

Fair Lab Record:

All Students attending the System Software and Microprocessors Lab should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, Aim of Experiment, Details of Experiment including algorithm and Result of Experiment. The left hand page should contain a print out of the code used for the experiment and sample output obtained for a set of input.

Syllabus**MICROPROCESSOR LAB**

- I. Assembly Language Programming Exercises/Experiments using 8086 Trainer kit
- II. Exercises/Experiments using MASM (PC required)
- III. Interfacing Exercises/Experiments with 8086 trainer kit through Assembly Language programming
- IV. Exercises/Experiments using 8051 trainer kit

SYSTEM SOFTWARE LAB:

- I. Experiments related to the operating system.
- II. Exercises/Experiments related to the assemblers, loaders and macroprocessors

Text Books

1. Bhurchandi and Ray, Advanced Microprocessors and Peripherals, Third Edition McGraw Hill.
2. Andrew S Tanenbaum, “Modern Operating Systems” , 4th Edition, Prentice Hall, 2015.
3. Leland L. Beck, System Software: An Introduction to Systems Programming, 3/E, Pearson Education Asia, 1997.

Reference Books

1. A. NagoorKani, Microprocessors and Microcontrollers, Second Edition, Tata McGraw Hill
2. Douglas V. Hall, SSSP Rao, Microprocessors and Interfacing, Third Edition, McGrawHill Education.
3. William Stallings, “Operating systems”, 6th Edition, Pearson, Global Edition, 2015.
4. Garry Nutt, Nabendu Chaki, Sarmistha Neogy, “Operating Systems”, 3rd Edition, Pearson Education.
5. D.M. Dhamdhare, Systems Programming and Operating Systems, Second Revised Edition, Tata McGraw Hill.

Practice Questions**MICROPROCESSORS LAB : List of Exercises/ Experiments**

(Minimum 10 Exercises (at least 2 questions from each part I, II, III & IV)) : 2 Hrs/week

I. Assembly Language Programming Exercises/Experiments using 8086 Trainer kit

1. Implementation of simple decimal arithmetic and bit manipulation operations.
2. Implementation of code conversion between BCD, Binary, Hexadecimal and ASCII.
3. Implementation of searching and sorting of 16-bit numbers.

II. Exercises/Experiments using MASM (PC Required)

4. Study of Assembler and Debugging commands.
5. Implementation of decimal arithmetic (16 and 32 bit) operations.
6. Implementation of String manipulations.
7. Implementation of searching and sorting of 16-bit numbers.

III. Interfacing Exercises/Experiments with 8086 trainer kit through Assembly Language Programming

8. Interfacing with stepper motor - Rotate through any given sequence.
9. Interfacing with 8255 (mode0 and mode1 only).
10. Interfacing with 8279 (Rolling message, 2 key lockout and N-key rollover implementation).

11. Interfacing with Digital-to-Analog Converter.

IV. Exercises/Experiments using 8051 trainer kit

12. Familiarization of 8051 trainer kit by executing simple Assembly Language programs such as decimal arithmetic and bit manipulation.
13. Implementation of Timer programming (in mode1).

SYSTEM SOFTWARE LAB: List of Exercises/ Experiments

(Minimum 8 Exercises (at least 3 and 5 questions from each part V and VI)) : 2

Hrs/week

V. Exercises/Experiments from operating system

1. Simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time.
 - a) FCFS b) SJF c) Round Robin (pre-emptive) d) Priority
2. Simulate the following file allocation strategies.
 - a) Sequential b) Indexed c) Linked
3. Implement the different paging techniques of memory management.
4. Simulate the following file organization techniques
 - a) Single level directory b) Two level directory c) Hierarchical
5. Implement the banker's algorithm for deadlock avoidance.
6. Simulate the following disk scheduling algorithms.
 - a) FCFS b) SCAN c) C-SCAN
7. Simulate the following page replacement algorithms:
 - a) FIFO b) LRU c) LFU

VI. Exercises/Experiments from assemblers, loaders and macroprocessor

1. Implement pass one of a two pass assembler.
2. Implement pass two of a two pass assembler.
3. Implement a single pass assembler.
4. Implement a two pass macro processor
5. Implement a single pass macro processor.
6. Implement an absolute loader.
7. Implement a relocating loader

CSL 333	DATABASE MANAGEMENT SYSTEMS LAB	Category	L	T	P	Credits	Year of introduction
		PCC	0	0	4	2	2019

Preamble:

The Database Management Systems course is intended to impart the elementary concepts of a database management system to students and equip them to design and implement a database application based on those concepts. This course helps the learners to get practical exposure on database creation, SQL queries creation, transaction processing and NoSQL & MongoDB based operations. The course enables the students to create, manage and administer the databases, develop necessary tools for the design and development of the databases, and to understand emerging technologies to handle Big Data.

Prerequisite: A sound knowledge of the basics of relational DBMS.

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Design database schema for a given real world problem-domain using standard design and modeling approaches. (Cognitive Knowledge Level: Apply)
CO2	Construct queries using SQL for database creation, interaction, modification, and updation. (Cognitive Knowledge Level: Apply)
CO3	Design and implement triggers and cursors. (Cognitive Knowledge Level: Apply)
CO4	Implement procedures, functions, and control structures using PL/SQL. (Cognitive Knowledge Level: Apply)
CO5	Perform CRUD operations in NoSQL Databases. (Cognitive Knowledge Level: Apply)
CO6	Develop database applications using front-end tools and back-end DBMS. (Cognitive Knowledge Level: Create)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	☑	☑	☑		☑			☑		☑		☑
CO2	☑	☑	☑		☑			☑		☑		☑
CO3	☑	☑	☑	☑	☑			☑		☑		☑
CO4	☑	☑	☑	☑	☑			☑		☑		☑
CO5	☑	☑	☑		☑			☑		☑		☑
CO6	☑	☑	☑	☑	☑	☑		☑	☑	☑	☑	☑

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern:

Bloom's Category	Continuous Assessment Test (Internal Exam) Percentage	End Semester Examination Percentage
Remember	20	20
Understand	20	20
Apply	60	60
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 15 marks
Continuous Evaluation in Lab	: 30 marks
Continuous Assessment Test	: 15 marks
Viva-voce	: 15 marks

Internal Examination Pattern : The marks will be distributed as Schema/Logic: 30 marks, Program/Queries: 20 marks, Output: 20 marks, and Viva: 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern:

The marks will be distributed as Schema/Logic: 30 marks, Program/Queries: 20 marks, Output: 20 marks, and Viva: 30 marks. Total 100 marks will be converted out of 75 for the End Semester Examination.

DBMS software: Oracle, MySQL, SQL Server, PostgreSQL, MongoDB.

Front end Tool: Java

Fair Lab Record:

All Students attending the DBMS Lab should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record, the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, Aim of Experiment, Schemas/Menu & Form Design, and Query questions. The left hand page should contain Queries and sample output(relations created, Form, and Menu Output) obtained for a set of input.

Syllabus

1. Design a database schema for an application with ER diagram from a problem description **.
2. Creation, modification, configuration, and deletion of databases using UI and SQL Commands **.
3. Creation of database schema - DDL (create tables, set constraints, enforce relationships, create indices, delete and modify tables). Export ER diagram from the database and verify relationships** (with the ER diagram designed in step 1).

4. Database initialization - Data insert, Data import to a database (bulk import using UI and SQL Commands)**.
5. Practice SQL commands for DML (insertion, updating, altering, deletion of data, and viewing/querying records based on condition in databases)**.
6. Implementation of built-in functions in RDBMS**.
7. Implementation of various aggregate functions in SQL**.
8. Implementation of Order By, Group By & Having clause **.
9. Implementation of set operators nested queries, and join queries **.
10. Implementation of queries using temp tables.
11. Practice of SQL TCL commands like Rollback, Commit, Savepoint **.
12. Practice of SQL DCL commands for granting and revoking user privileges **.
13. Practice of SQL commands for creation of views and assertions **.
14. Implementation of various control structures like IF-THEN, IF-THEN-ELSE, IF-THEN-ELSIF, CASE, WHILE using PL/SQL **.
15. Creation of Procedures, Triggers and Functions**.
16. Creation of Packages **.
17. Creation of Cursors **.
18. Creation of PL/SQL blocks for exception handling **.
19. Database backup and restore using commands.
20. Query analysis using Query Plan/Show Plan.
21. Familiarization of NoSQL Databases and CRUD operations**.
22. Design a database application using any front end tool for any problem selected. The application constructed should have five or more tables**.

** mandatory

Text Books

1. Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education, 2013.
2. Sliberschatz A., H. F. Korth and S. Sudarshan, Database System Concepts, 6/e, McGraw Hill, 2011.

References

1. Adam Fowler, NoSQL for Dummies, John Wiley & Sons, 2015
2. NoSQL Data Models: Trends and Challenges (Computer Engineering: Databases and Big Data), Wiley, 2018

Practice Questions

Design a normalized database schema for the following requirement.

The requirement: A library wants to maintain the record of books, members, book issue, book return, and fines collected for late returns, in a database. The database can be loaded with book information. Students can register with the library to be a member. Books can be issued to students with a valid library membership. A student can keep an issued book with him/her for a maximum period of two weeks from the date of issue, beyond which a fine will be charged. Fine is calculated based on the delay in days of return. For 0-7 days: Rs 10, For 7 – 30 days: Rs 100, and for days above 30 days: Rs 10 will be charged per day.

Sample Database Design

BOOK (**Book_Id**, Title, Language_Id, MRP, Publisher_Id, Published_Date, Volume, Status) // Language_Id, Publisher_Id are FK (Foreign Key)

AUTHOR(Author_Id, Name, Email, Phone_Number, Status)

BOOK_AUTHOR(Book_Id, Author_Id) // many-to-many relationship, both columns are PKFK (Primary Key and Foreign Key)

PUBLISHER(Publisher_id, Name, Address)

MEMBER(Member_Id, Name, Branch_Code, Roll_Number, Phone_Number, Email_Id, Date_of_Join, Status)

BOOK_ISSUE(Issue_Id, Date_Of_Issue, Book_Id, Member_Id, Expected_Date_Of_Return, Status) // Book+Id and Member_Id are FKs

BOOK_RETURN(Issue_Id, Actual_Date_Of_Return, LateDays, LateFee) // Issue_Id is PK and FK

LANGUAGE(Language_id, Name) //Static Table for storing permanent data

LATE_FEE_RULE(FromDays, ToDays, Amount) // Composite Key

EXERCISES

1. Create a normalized database design with proper tables, columns, column types, and constraints
2. Create an ER diagram for the above database design.
3. Write SQL commands to
 - a. Create a database by name *Library*. Drop the database and re-create it.
 - b. Create DDL statements and create the tables and constraints (from the design) in the database created in step-a (*Library*)

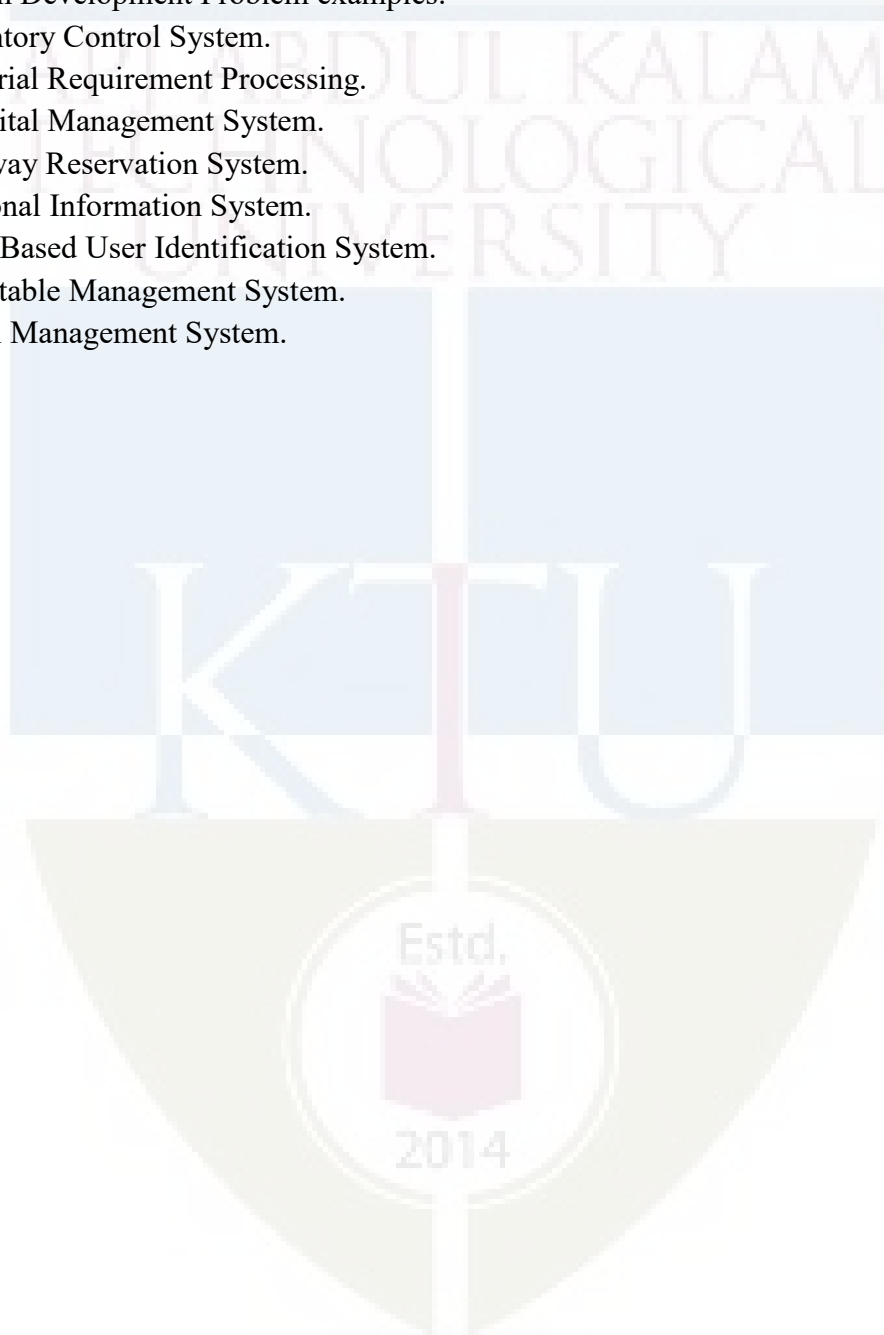
Notes: [Create a script file and execute it. Create the script file in such a way that,,if the table exists, drop the tables and recreate)]

- c. Create and execute DROP TABLE command in tables with and without FOREIGN KEY constraints.
 - d. Create and execute ALTER TABLE command in tables with data and without data.
 - e. Create and execute SQL commands to build indices on Member_Id and Book_Id on table Book_Issue.
 - f. Create and execute GRANT/REVOKE commands on tables.
 - g. Create and execute SQL commands to insert data into each of the tables designed
 - h. Learn and execute bulk import of data to tables from CSV files (insert 1000 records of books into the BOOK table from a CSV file).
 - i. Create and execute UPDATE/DELETE commands on tables. Try to update/delete rows with Primary and Foreign Keys. Try bulk updates or deletes using SQL UPDATE statement
4. Write SQLQuery to retrieve the following information
 - a. Get the number of books written by a given author
 - b. Get the list of publishers and the number of books published by each publisher
 - c. Get the names of authors who jointly wrote more than one book.
 - d. Get the list of books that are issued but not returned
 - e. Get the list of students who reads only 'Malayalam' books
 - f. Get the total fine collected for the current month and current quarter
 - g. Get the list of students who have overdue (not returned the books even on due date)
 - h. Calculate the fine (as of today) to be collected from each overdue book.
 - i. Members who joined after Jan 1 2021 but has not taken any books
 5. Book return should insert an entry into the Book_Return table and also update the status in Book_Issue table as 'Returned'. Create a database *TRANSACTION* to do this operation (stored procedure).
 6. Create a database view 'Available_Books', which will list out books that are currently available in the library
 7. Create a database procedure to add, update and delete a book to the Library database (use parameters).
 8. Use cursors and create a procedure to print Books Issue Register (page wise – 20 rows in a page)
 9. Create a history table (you may use the same structure without any keys) for the MEMBER table and copy the original values of the row being updated to the history table using a TRIGGER.
 10. NoSQL Exercise
 - a. Practice Mongo DB CRUD operations. Refer:
<https://docs.mongodb.com/manual/crud/>

- b. You may use a MongoDB local installation or cloud MongoDB services like MongoDB Atlas for this exercise
- c. For documentation: Refer: <https://docs.mongodb.com/manual/introduction/>

11. Application Development Problem examples:

- 1) Inventory Control System.
- 2) Material Requirement Processing.
- 3) Hospital Management System.
- 4) Railway Reservation System.
- 5) Personal Information System.
- 6) Web Based User Identification System.
- 7) Timetable Management System.
- 8) Hotel Management System.



APJ ABDUL KALAM
TECHNOLOGICAL
UNIVERSITY

SEMESTER V

MINOR



CST 381	CONCEPTS IN SOFTWARE ENGINEERING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0		

Preamble: This course provides fundamental knowledge in the Software Development Process. It covers Software Development, Quality Assurance and Project Management concepts. This course enables the learners to apply state of the art industry practices in Software development.

Prerequisite: Basic understanding of Object Oriented Design and Development.

Course Outcomes: After the completion of the course the student will be able to

CO1	Differentiate Traditional and Agile Software Development approaches (Cognitive Knowledge Level: Understand)
CO2	Prepare Software Requirement Specification and Software Design for a given problem. (Cognitive Knowledge Level: Apply)
CO3	Justify the significance of design patterns and licensing terms in software development, prepare testing, maintenance and DevOps strategies for a project. (Cognitive Knowledge Level: Apply)
CO4	Make use of software project management concepts while planning, estimation, scheduling, tracking and change management of a project, with proper application of SCRUM, Kanban and Lean frameworks. (Cognitive Knowledge Level: Apply)
CO5	Utilize SQA practices, Process Improvement techniques and Technology improvements namely cloud based software model and containers & microservices in a Software Development Process. (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	☑	☑	☑	☑		☑						☑
CO2	☑	☑	☑	☑		☑				☑	☑	☑

CO3	✓	✓	✓	✓				✓		✓	✓	✓
CO4	✓	✓	✓	✓		✓			✓	✓	✓	✓
CO5	✓	✓	✓	✓		✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	30	30	30

Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Tests : **25 marks**

Continuous Assessment Assignment : **15 marks** (Each student shall identify a software development problem and prepare Requirements Specification, Design Document, Project Plan and Test case documents for the identified problem as the assignment.)

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks.

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 subdivisions and carries 14 marks.

Syllabus**Module 1 : Introduction to Software Engineering (8 hours)**

Introduction to Software Engineering - Professional software development, Software engineering ethics. Software process models - The waterfall model, Incremental development. Process activities - Software specification, Software design and implementation, Software validation, Software evolution. Coping with change - Prototyping, Incremental delivery, Boehm's Spiral Model. Agile software development - Agile methods, agile manifesto - values and principles. Agile development techniques, Agile Project Management. Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care.

Module 2 : Requirement Analysis and Design (10 hours)

Functional and non-functional requirements, Requirements engineering processes. Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix. Developing use cases, Software Requirements Specification Template, Personas, Scenarios, User stories, Feature identification. Design concepts - Design within the context of software engineering, Design Process, Design concepts, Design Model. Architectural Design - Software Architecture, Architectural Styles, Architectural considerations, Architectural Design Component level design - What is a component?, Designing Class-Based Components, Conducting Component level design, Component level design for web-apps. Template of a Design Document as per "IEEE Std 1016-2009 IEEE Standard for Information Technology Systems Design Software Design Descriptions". Case study: The Ariane 5 launcher failure.

Module 3 : Implementation and Testing (12 hours)

Object-oriented design using the UML, Design patterns, Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD. Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. Informal Review, Formal Technical Reviews, Post-mortem evaluations. Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing, Debugging, White box testing, Path testing, Control Structure testing, Black box testing, Testing Documentation and Help facilities. Test automation, Test-driven development, Security testing. Overview of DevOps and Code Management - Code management, DevOps automation, CI/CD/CD. Software Evolution - Evolution processes, Software maintenance.

Module 4 : Software Project Management (8 hours)

Software Project Management - Risk management, Managing people, Teamwork. Project Planning, Software pricing, Plan-driven development, Project scheduling, Agile planning. Estimation techniques, COCOMO cost modeling. Configuration management, Version management, System building, Change management, Release management, Agile software management - SCRUM framework. Kanban methodology and lean approaches.

Module 5 : Software Quality and Process Improvement (6 hours)

Software Quality, Software Quality Dilemma, Achieving Software Quality Elements of Software Quality Assurance, SQA Tasks , Software measurement and metrics. Software Process Improvement(SPI), SPI Process CMMI process improvement framework, ISO 9001:2000 for Software.

Text Books

1. Book 1 - Ian Sommerville, Software Engineering, Pearson Education, Tenth edition, 2015.
2. Book 2 - Roger S. Pressman, Software Engineering : A practitioner's approach, McGraw Hill publication, Eighth edition, 2014
3. Book 3 - Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson Education, First Edition, 2020.

References

1. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
2. IEEE Std 1016-2009 IEEE Standard for Information Technology—Systems Design—Software Design Descriptions
3. David J. Anderson, Kanban, Blue Hole Press 2010
4. David J. Anderson, Agile Management for Software Engineering, Pearson, 2003
5. Walker Royce, Software Project Management : A unified framework, Pearson Education, 1998
6. Steve. Denning, The age of agile, how smart companies are transforming the way work gets done. New York, Amacom, 2018.
7. Satya Nadella, Hit Refresh: The Quest to Rediscover Microsoft's Soul and Imagine a Better Future for Everyone, Harper Business, 2017
8. Henrico Dolfing, Project Failure Case Studies: Lessons learned from other people's mistakes, Kindle edition
9. Mary Poppendieck, Implementing Lean Software Development: From Concept to Cash, Addison-Wesley Signature Series, 2006
10. StarUML documentation - <https://docs.staruml.io/>
11. OpenProject documentation - <https://docs.openproject.org/>

12. BugZilla documentation - <https://www.bugzilla.org/docs/>
13. GitHub documentation - <https://guides.github.com/>
14. Jira documentation - <https://www.atlassian.com/software/jira>

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are the advantages of an incremental development model over a waterfall model?
2. Compare agile software development with traditional software development?

Course Outcome 2 (CO2):

1. How to prepare a software requirement specification?
2. Differentiate between Architectural design and Component level design.
3. How do agile approaches help software developers to capture and define the user requirements effectively?
4. What is the relevance of the SRS specification in software development?
5. Prepare a use case diagram for a library management system.

Course Outcome 3 (CO3):

1. Differentiate between the different types of software testing strategies.
2. What are the benefits of DevOps?
3. How do design patterns help software architects communicate the design of a complex system effectively?
4. What are the proactive approaches one can take to optimise efforts in the testing phase?

Course Outcome 4 (CO4):

1. What are the activities involved in software project management?
2. What is the need for SCRUM, Kanban and Lean methodologies?
3. What are the benefits of rolling level planning in software project management and how would you implement it?
4. How would you assess the risks in your software development project? How would you plan for risk mitigation and contingency?

Course Outcome 5 (CO5):

1. What is the importance of Software Process improvement?
2. How will retrospectives help in improving the software development process?
3. What are the important skills required for the SQA role?
4. How would you use project history data as a prediction tool to plan future projects?

Model Question Paper

QP CODE:

Reg No: _____

Name : _____

PAGES : 3

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FIFTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR**

Course Code: CST 381

Course Name: Concepts in Software Engineering

Duration: 3 Hrs

Max. Marks : 100

PART A

Answer all Questions. Each question carries 3 Marks

1. Explain why professional software that is developed for a customer is not simply the programs that have been developed and delivered
2. Incremental software development could be very effectively used for customers who do not have a clear idea about the systems needed for their operations. Discuss.
3. Identify and briefly describe four types of requirements that may be defined for a computer based system.
4. Describe software architecture in your own words.
5. What are the major differences between GPL and LGPL?
6. Compare between white box testing and black box testing.
7. What is the importance of risk management in software project management?
8. Explain COCOMO cost estimation model
9. Describe the software quality dilemma in your own words
10. Which are the levels of the CMMI model?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 marks)

11. (a) Compare between waterfall model and spiral model (8)
- (b) Explain Agile methods and Agile manifesto (6)
- OR**
12. (a) Explain software process activities (7)
- (b) Explain Agile Development techniques and Agile Project Management. (7)
13. (a) What are functional and nonfunctional requirements? Imagine that you are developing a library management software for your college, identify at least 8 functional requirements and 4 nonfunctional requirements. (10)
- (b) What are the contents of a software requirement specification? (4)
- OR**
14. (a) Explain Personas, Scenarios, User stories and Feature identification? (8)
- (b) Compare between Software Architecture design and Component level design (6)
15. (a) Describe the formal and informal review techniques in detail. (6)
- (b) Explain various software testing strategies. (8)
- OR**
16. (a) Explain DevOps CI/CD/CD in detail. (8)
- (b) Explain test driven development. (6)
17. (a) What is a critical path and demonstrate its significance in a project schedule with the help of a sample project schedule. (6)
- (b) Explain plan driven development and project scheduling (6)

OR

18. (a) Explain the SCRUM framework. (8)
- (b) What is algorithmic cost modeling? What problems does it suffer from when compared with other approaches to cost estimation? (6)
19. (a) Explain elements of Software Quality Assurance and SQA Tasks. (8)
- (b) Explain the SPI process. (6)

OR

20. (a) Compare between CMMI and ISO 9001:2000 (8)
- (b) Compare Quality Control and Quality Assurance. (6)

Teaching Plan [44 hours]		
Module 1 : Introduction to Software Engineering (8 hours)		Hours
1.1	Introduction to Software Engineering. [Book 1, Chapter 1]	1 hour
1.2	Software process models [Book 1 - Chapter 2]	1 hour
1.3	Process activities [Book 1 - Chapter 2]	1 hour
1.4	Coping with change [Book 1 - Chapter 2, Book 2 - Chapter 4]	1 hour
1.5	Agile software development [Book 1 - Chapter 3]	1 hour
1.6	Agile development techniques [Book 1 - Chapter 3]	1 hour
1.7	Agile Project Management.[Book 1 - Chapter 3]	1 hour
1.8	Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care. [Book 1 - Chapter 1]	1 hour
Module 2 : Requirement Analysis and Design (10 hours)		
2.1	Functional and non-functional requirements, Requirements engineering processes [Book 1 - Chapter 4]	1 hour

2.2	Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix [Book 1 - Chapter 4]	1 hour
2.3	Developing use cases, Software Requirements Specification Template [Book 2 - Chapter 8]	1 hour
2.4	Personas, Scenarios [Book 3 - Chapter 3]	1 hour
2.5	User stories, Feature identification [Book 3 - Chapter 3]	1 hour
2.6	Design concepts [Book 2 - Chapter 12]	1 hour
2.7	Architectural Design [Book 2 - Chapter 13]	1 hour
2.8	Component level design [Book 2 - Chapter 14]	1 hour
2.9	Component level design, Design Document Template. [Book 2 - Chapter 14, Ref - 2]	1 hour
2.10	Case study: The Ariane 5 launcher failure. [Book 2 - Chapter 16]	1 hour
Module 3 : Implementation and Testing (12 hours)		
3.1	Object-oriented design using the UML, Design patterns [Book 1 - Chapter 7]	1 hour
3.2	Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD [Book 1 - Chapter 7]	1 hour
3.3	Review Techniques - Cost impact of Software Defects, Code review. [Book 2 - Chapter 20]	1 hour
3.4	Informal Review, Formal Technical Reviews, Post-mortem evaluations. [Book 2 - Chapter 20]	1 hour
3.5	Software testing strategies [Book 2 - Chapter 22]	1 hour
3.6	Software testing strategies [Book 2 - Chapter 22]	1 hour
3.7	White box testing, Path testing, Control Structure testing [Book 2 - Chapter 23]	1 hour
3.8	Black box testing. Test documentation [Book 2 - Chapter 23]	1 hour
3.9	Test automation, Test-driven development [Book 3 - Chapter 9]	1 hour
3.10	Security testing. DevOps and Code Management [Book 3 - Chapter 9, Chapter 10]	1 hour
3.11	DevOps and Code Management - Code management, DevOps automation, CI/CD/CD. [Book 3 - Chapter 10]	1 hour

3.12	Software Evolution - Evolution processes, Software maintenance. [Book 1 - Chapter 9]	1 hour
Module 4 : Software Project Management (8 hours)		
4.1	Software Project Management - Risk management, Managing people, Teamwork [Book 1 - Chapter 22]	1 hour
4.2	Project Planning - Software pricing, Plan-driven development, Project scheduling, Agile planning [Book 1 - Chapter 23]	1 hour
4.3	Estimation techniques [Book 1 - Chapter 23]	1 hour
4.4	Configuration management [Book 1 - Chapter 25]	1 hour
4.5	Agile software management - SCRUM framework [Book 2 - Chapter 5]	1 hour
4.6	Agile software management - SCRUM framework [Book 2 - Chapter 5]	1 hour
4.7	Kanban methodology and lean approaches. [Ref 9 - Chapter 2]	1 hour
4.8	Kanban methodology and lean approaches.[Ref 9 - Chapter 2]	1 hour
Module 5 : Software Quality, Process Improvement and Technology trends (6 hours)		
5.1	Software Quality, Software Quality Dilemma, Achieving Software Quality. [Book 2 - Chapter 19]	1 hour
5.2	Elements of Software Quality Assurance, SQA Tasks [Book 3 - Chapter 21]	1 hour
5.3	Software measurement and metrics. [Book 3 - Chapter 21]	1 hour
5.4	Software Process Improvement(SPI), SPI Process[Book 2 - Chapter 37]	1 hour
5.5	Software Process Improvement(SPI), SPI Process[Book 2 - Chapter 37]	1 hour
5.6	CMMI process improvement framework, ISO 9001:2000 for Software. [Book 2 - Chapter 37]	1 hour

CST 383	CONCEPTS IN MACHINE LEARNING	Category	L	T	P	Credit	Year of introduction
		VAC	3	1	0	4	2019

Preamble: This course enables the learners to understand the fundamental concepts and algorithms in machine learning. The course covers the standard and most popular supervised learning algorithms such as linear regression, logistic regression, decision trees, Bayesian learning & the naive Bayes algorithm, support vector machines & kernels, basic clustering algorithms and dimensionality reduction methods. This course helps the students to provide machine learning based solutions to real world problems.

Prerequisite: Familiarity with basics in linear algebra, probability and Python programming.

Course Outcomes	
CO1	Illustrate Machine Learning concepts and basic parameter estimation methods. (Cognitive Knowledge Level: Apply)
CO2	Demonstrate supervised learning concepts (regression, linear classification). (Cognitive Knowledge Level: Apply)
CO3	Illustrate the concepts of Multilayer neural network and Support Vector Machine (Cognitive Knowledge Level: Apply)
CO4	Describe unsupervised learning concepts and dimensionality reduction techniques. (Cognitive Knowledge Level: Apply)
CO5	Solve real life problems using appropriate machine learning models and evaluate the performance measures (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓	✓							✓
CO2	✓	✓	✓	✓	✓							✓
CO3	✓	✓	✓	✓	✓							✓

CO4	✓	✓	✓	✓	✓							✓
CO5	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks
 Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

Module-1 (Overview of machine learning)

Machine learning paradigms-supervised, semi-supervised, unsupervised, reinforcement learning. Basics of parameter estimation - maximum likelihood estimation(MLE) and maximum a posteriori estimation(MAP). Introduction to Bayesian formulation.

Module-2 (Supervised Learning)

Regression - Linear regression with one variable, Linear regression with multiple variables, solution using gradient descent algorithm and matrix method, basic idea of overfitting in regression. Linear Methods for Classification- Logistic regression, Perceptron, Naive Bayes, Decision tree algorithm ID3.

Module-3 (Neural Networks (NN) and Support Vector Machines (SVM))

NN - Multilayer feed forward network, Activation functions (Sigmoid, ReLU, Tanh), Backpropagation algorithm.

SVM - Introduction, Maximum Margin Classification, Mathematics behind Maximum Margin Classification, Maximum Margin linear separators, soft margin SVM classifier, non-linear SVM, Kernels for learning non-linear functions, polynomial kernel, Radial Basis Function(RBF).

Module-4 (Unsupervised Learning)

Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitioned clustering, Expectation maximization (EM) for soft clustering. Dimensionality reduction – Principal Component Analysis, factor Analysis, Multidimensional scaling, Linear Discriminant Analysis.

Module-5 (Classification Assessment)

Classification Performance measures - Precision, Recall, Accuracy, F-Measure, Receiver Operating Characteristic Curve(ROC), Area Under Curve(AUC). Bootstrapping, Cross Validation, Ensemble methods, Bias-Variance decomposition. Case Study: Develop a classifier for face detection.

Text Book

1. Ethem Alpaydin, Introduction to Machine Learning, 2nd edition, MIT Press 2010.
2. Mohammed J. Zaki and Wagner Meira, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, First South Asia edition, 2016.
3. Jake VanderPlas, Python Data Science Handbook, O'Reilly Media, 2016
4. Tom Mitchell, Machine Learning, McGraw-Hill, 1997.

Reference Books

1. Christopher Bishop. Neural Networks for Pattern Recognition, Oxford University Press, 1995.
2. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective, MIT Press 2012.
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements Of Statistical Learning, Second edition Springer 2007.
4. P. Langley, Elements of Machine Learning, Morgan Kaufmann, 1995.
5. Richert and Coelho, Building Machine Learning Systems with Python.
6. Davy Cielen, Arno DB Meysman and Mohamed Ali. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Dreamtech Press 2016.

Sample Course Level Assessment Questions

Course Outcome1(CO1):

1. A coin is tossed 100 times and lands heads 62 times. What is the maximum likelihood estimate for θ , the probability of heads.
2. Suppose data x_1, \dots, x_n are independent and identically distributed drawn from an exponential distribution $exp(\lambda)$. Find the maximum likelihood for λ .
3. Suppose x_1, \dots, x_n are independent and identically distributed(iid) samples from a distribution with density

$$f_X(x|\theta) = \begin{cases} \frac{\theta x^{\theta-1}}{3^\theta}, & 0 \leq x \leq 3 \\ 0, & \text{otherwise} \end{cases}$$

Find the maximum likelihood estimate(MLE) for θ .

4. Find the maximum likelihood estimator (MLE) and maximum a posteriori (MAP) estimator for the mean of a univariate normal distribution. Assume that we have N samples, x_1, \dots, x_N independently drawn from a normal distribution with known variance σ^2 and unknown mean μ and the prior distribution for the mean is itself a normal distribution with mean ν and variance β^2 . What happens to the MLE and MAP estimators as the number of samples goes to infinity.

Course Outcome 2 (CO2):

1. Explain the difference between (batch) gradient descent and stochastic gradient descent. Give an example of when you might prefer one over the other.
2. Suppose that you are asked to perform linear regression to learn the function that outputs y , given the D -dimensional input x . You are given N independent data points, and that all the D attributes are linearly independent. Assuming that D is around 100, would you prefer the closed form solution or gradient descent to estimate the regressor?
3. Suppose you have a three class problem where class label $y \in \{0, 1, 2\}$ and each training example X has 3 binary attributes $X_1, X_2, X_3 \in \{0, 1\}$. How many parameters (probability distribution) do you need to know to classify an example using the Naive Bayes classifier?

Course Outcome 3 (CO3):

1. What are support vectors and list any three properties of the support vector classifier solution?
2. Why do you use kernels to model a projection from attributes into a feature space, instead of simply projecting the dataset directly?
3. Describe how Support Vector Machines can be extended to make use of kernels. Illustrate with reference to the Gaussian kernel $K(x, y) = e^{-z}$, where $z = (x-y)^2$.

4. Briefly explain one way in which using tanh instead of logistic activations makes optimization easier.
5. ReLU activation functions are most used in neural networks instead of the tanh activation function. Draw both activation functions and give a) an advantage of the ReLU function compared to the tanh function. b) a disadvantage of the ReLU function compared to the tanh function.

Course Outcome 4(CO4):

1. Describe cluster analysis? Identify two applications where cluster analysis can be applied to multimedia data?
2. Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):
 - (i) Compute the Euclidean distance between the two objects.
 - (ii) Compute the Manhattan distance between the two objects.
3. Use PCA to reduce the dimension from 2 to 1 for the design matrix X .

$$X = \begin{bmatrix} 6 & -4 \\ -3 & 5 \\ -2 & 6 \\ 7 & -3 \end{bmatrix}$$

4. What is Principal Component Analysis (PCA)? Which eigen value indicates the direction of largest variance?
5. Suppose that one runs a principal component analysis on a data set and tells that the percentage of variance explained by the first 3 components is 80%. How is this percentage of variance explained?

Course Outcome 5 (CO5):

1. Suppose that you are contacted by a food processing company that wants you to develop a classifier that detects whether a rat is present in an image. You collect a large dataset of images by crawling the web, and have annotators determine which images contain rats. This set of images can then be used as the training set for your classifier.
 - a. Suggest a machine learning method to use for this classification task and evaluate its performance.
 - b. After you have delivered your solution to the company, they get back to you and complain that when they evaluate on a new test set, they get precision and recall values that are much lower than what you reported to them. Explain what might have gone wrong and propose remedial measures .
2. A real estate firm would like to build a system that predicts the sale prices of a house. They create a spreadsheet containing information about 1,500 house sales in the Kochi

area. In addition to the price, there are 10 features describing the house, such as number of bedrooms, total indoor area, lot area, a swimming pool, location, etc. Explain how you would implement a machine learning model that would solve this prediction task. Give all steps you would carry out when developing it. Explain why the model you built is probably useless in the long run.

3. For a classifier, the confusion matrix is given by:

	+	-
+	9	9
-	1	5

What is the precision, recall and accuracy of that classifier?

Model Question Paper

QP CODE:

PAGES:3

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FIFTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH &
YEAR

Course Code: CST 383

Course Name: CONCEPTS IN MACHINE LEARNING

Max.Marks:100

Duration: 3

Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Define supervised learning? Name special cases of supervised learning depending on whether the inputs/outputs are categorical, or continuous.
2. Differentiate between Maximum Likelihood estimation (MLE) and Maximum a Posteriori (MAP) estimation?
3. What is overfitting and why is it a problem? Give an example of a method to reduce the risk of overfitting.
4. Specify the basic principle of gradient descent algorithm.
5. Suppose that you have a linear support vector machine(SVM) binary classifier. Consider a point that is currently classified correctly, and is far away from the decision boundary. If you

remove the point from the training set, and re-train the classifier, will the decision boundary change or stay the same? Justify your answer.

6. Mention the primary motivation for using the kernel trick in machine learning algorithms?
7. Expectation maximization (EM) is designed to find a maximum likelihood setting of the parameters of a model when some of the data is missing. Does the algorithm converge? If so, do you obtain a locally or globally optimal set of parameters?
8. Illustrate the strength and weakness of k-means algorithm.
9. Classifier A attains 100% accuracy on the training set and 70% accuracy on the test set. Classifier B attains 70% accuracy on the training set and 75% accuracy on the test set. Which one is a better classifier. Justify your answer.
10. How does bias and variance trade-off affect machine learning algorithms?

(10x3=30)

Part B

Answer any one Question from each module. Each question carries 14 Marks

11. a) Suppose that X is a discrete random variable with the following probability mass function: where $0 \leq \theta \leq 1$ is a parameter. The following 10 independent observations

X	0	1	2	3
$P(X)$	$2\theta/3$	$\theta/3$	$2(1 - \theta)/3$	$(1 - \theta)/3$

were taken from such a distribution: $(3, 0, 2, 1, 3, 2, 1, 0, 2, 1)$. What is the maximum likelihood estimate of θ . (6)

- b) A gamma distribution with parameters α, β has the following density function, where $\Gamma(t)$ is the gamma function.

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

If the posterior distribution is in the same family as the prior distribution, then we say that the prior distribution is the conjugate prior for the likelihood function. Using the Gamma distribution as a prior, show that the Exponential distribution is a conjugate prior of the Gamma distribution. Also, find the maximum a posteriori estimator for the parameter of the Exponential distribution as a function of α and β . (8)

OR

12. a) Traffic between 8AM and 9AM at a certain place was measured by counting the number of vehicles that passed at that time. Suppose the counts follow a Poisson process. A random sample of 9 observations was collected, having observed the following number of vehicles: $(95, 100, 80, 70, 110, 98, 97, 90, 70)$. Derive the maximum likelihood estimator for the

average number of vehicles that pass by that place between 8 AM and 9 AM, and compute the corresponding estimate using the given sample. (7)

b) Find the maximum a posteriori (MAP) estimator for the mean of a univariate normal distribution. Assume that we have N samples, x_1, \dots, x_N independently drawn from a normal distribution with known variance σ^2 and unknown mean μ and the prior distribution for the mean is itself a normal distribution with mean ν and variance β^2 . (7)

13.a) Derive the gradient descent training rule assuming for the target function $o_d = w_0 + w_1x_1 + \dots + w_nx_n$. Define explicitly the squared cost/error function E , assuming that a set of training examples D is provided, where each training example $d \in D$ is associated with the target output t_d . (10)

b) How can we interpret the output of a two-class logistic regression classifier as a probability? (4)

OR

14. a) In a two-class logistic regression model, the weight vector $w = [4, 3, 2, 1, 0]$. We apply it to some object that we would like to classify; the vectorized feature representation of this object is $x = [-2, 0, -3, 0.5, 3]$. What is the probability, according to the model, that this instance belongs to the positive class? (6)

b) The following dataset can be used to train a classifier that determines whether a given person is likely to own a car or not. There are three features: education level (primary, secondary, or university); residence (city or country); gender (female, male).

education	residence	gender	has car?
sec	country	female	yes
univ	country	female	yes
prim	city	male	no
univ	city	male	no
sec	city	female	no
sec	country	male	yes
prim	country	female	yes
univ	country	male	yes
sec	city	male	yes
prim	city	female	no
univ	city	female	no
prim	country	male	yes

Find the root attribute and justify your answer (8)

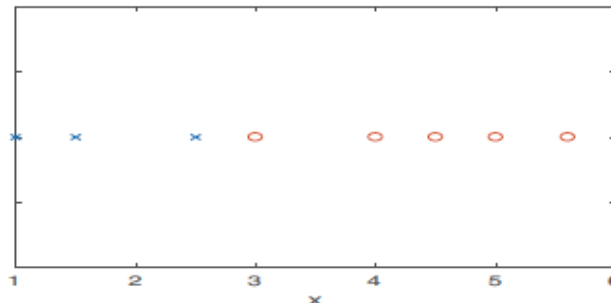
15. a) Consider a support vector machine whose input space is 2-D, and the inner products are computed by means of the kernel $K(x, y) = (x \cdot y + 1)^2 - 1$, where $x \cdot y$ denotes the ordinary inner product. Show that the mapping to feature space that is implicitly defined by this kernel is the mapping to 5-D given by (10)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \\ \sqrt{2} x_1 \\ \sqrt{2} x_2 \end{bmatrix}$$

- b) What is the basic idea of a Support Vector Machine? (4)

OR

16. a) Explain how back propagation can be used to solve XOR problem which is not linearly separable. (8)
- b) Consider the following one dimensional training data set, 'x' denotes negative examples and 'o' positive examples. The exact data points and their labels are given in the table. Suppose a SVM is used to classify this data. Indicate which are the support vectors and mark the decision boundary. Find the equation of the hyperplane. (6)



x	1	1.5	2.5	3	4	4.5	5	5.6
y	-1	-1	-1	1	1	1	1	1

17. a) Suppose that we have the following data (one variable). Use single linkage Agglomerative clustering to identify the clusters.
Daa: (2, 5, 9, 15, 16, 18, 25, 33, 33, 45). (8)

- b) Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):
- Compute the Euclidean distance between the two objects.
 - Compute the Manhattan distance between the two objects.

(iii) Compute the Minkowski distance between the two objects, using $p = 3$ (6)

OR

18. a) Suppose that we have the following data:

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>
(2,0)	(1,2)	(2,2)	(3,2)	(2,3)	(3,3)	(2,4)	(3,4)	(4,4)	(3,5)

Identify the cluster by applying the k-means algorithm, with $k = 2$. Try using initial cluster centers as far apart as possible. (10)

b) List the steps involved in Principal Component Analysis. (4)

19. a) Suppose the dataset had 9700 cancer-free images from 10000 images from cancer patients. Find precision, recall and accuracy? Is it a good classifier? Justify. (8)

Actual Class\ Predicted class	cancer = yes	cancer = no	Total
cancer = yes	90	210	300
cancer = no	140	9560	9700
Total	230	9770	10000

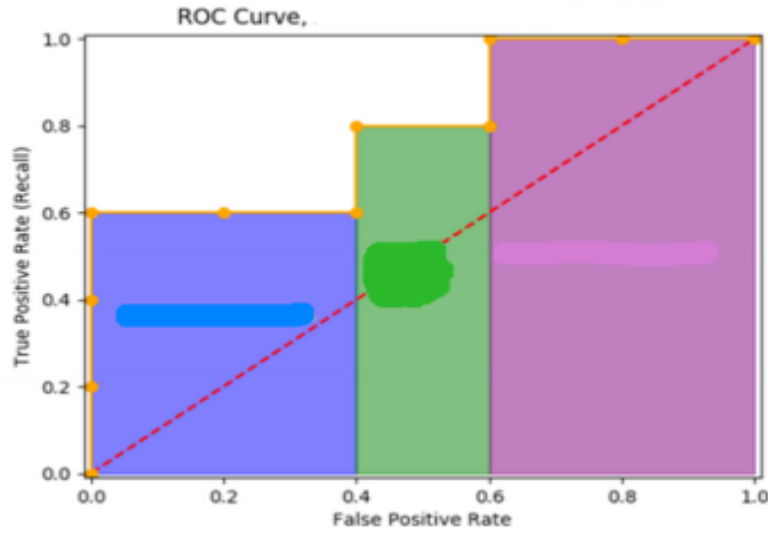
b) Suppose that you have a classification problem where our feature representation contains about 10,000,000 features. We would like to develop a classifier that can be deployed in a mobile phone, so preferably it should have a small memory footprint. Discuss one solution for how this can be done. (6)

OR

20. a) What are ROC space and ROC curve in machine learning? In ROC space, which points correspond to perfect prediction, always positive prediction and always negative prediction? Why? (6)

b) Suppose there are three classifiers A, B and C. The (FPR, TPR) measures of the three classifiers are as follows – A (0, 1), B (1, 1), C (1, 0.5). Which can be considered as a perfect classifier? Justify your answer. (4)

c) Given the following ROC Curve? Find the AUC? (4)



Teaching Plan

No	Contents	No of Lecture Hrs
Module 1 :Overview of machine learning (7 hours)		
1.1	Supervised, semi-supervised, unsupervised learning, reinforcement learning (Text Book (TB) 1: Chapter 1)	1hour
1.2	Maximum likelihood estimation(MLE) (TB 1: Section 4.2)	1hour
1.3	Maximum likelihood estimation (MLE)- example (TB 1: Section 4.2)	1hour
1.4	Maximum a posteriori estimation(MAP) (TB 4: Section 6.2)	1hour
1.5	Maximum a posteriori estimation(MAP)-example (TB 4: Section 6.2)	1hour
1.6	Bayesian formulation (TB 1: Section 14.1, 14.2)	1hour
1.7	Bayesian formulation -example (TB 1: Section 14.1, 14.2)	1hour
Module 2 : Supervised Learning (8 hours)		

2.1	Linear regression with one variable (TB 1: Section 2.6)	1 hour
2.2	Multiple variables, Solution using gradient descent algorithm and matrix method (No derivation required) (TB 1: Section 5.8)	1 hour
2.3	Overfitting in regression, Lasso and Ridge regularization	1 hour
2.4	Logistic regression	1 hour
2.5	Perceptron	1 hour
2.6	Naive Bayes (TB 2: Section 18.2)	1 hour
2.7	Decision trees (TB 2: Chapter 19)	1 hour
2.8	Decision trees- ID3 algorithm (TB 2: Chapter 19)	1 hour
Module 3 : Neural Networks and Support Vector Machines (TB 2: Chapter 21) (11 hours)		
3.1	Multilayer Feed forward Network, Activation Functions (Sigmoid, ReLU, Tanh)	1 hour
3.2	Back Propagation Algorithm	1 hour
3.3	Illustrative Example for Back Propagation	1 hour
3.4	Introduction, Maximum Margin Hyperplane,	1 hour
3.5	Mathematics behind Maximum Margin Classification	1 hour
3.6	Formulation of maximum margin hyperplane and solution	1 hour
3.7	Soft margin SVM	1 hour
3.8	Solution of Soft margin SVM	1 hour
3.9	Non-linear SVM	1 hour
3.10	Kernels for learning non-linear functions and properties of kernel functions.	1 hour
3.11	Example Kernels functions- Linear, RBF, Polynomial.	1 hour
Module 4 : Unsupervised Learning (10 hours)		
4.1	Similarity measures- Minkowski distance measures(Manhattan, Euclidean), Cosine Similarity	1 hour
4.2	Clustering - Hierarchical Clustering (TB 2: Chapter 14)	1 hour
4.3	K-means partitional clustering (TB 2: Chapter 13)	1 hour
4.4	Expectation maximization (EM) for soft clustering (TB 2: Chapter 13)	1 hour
4.5	Expectation maximization (EM) for soft clustering (TB 2: Chapter 13)	1 hour

4.6	Dimensionality reduction – Principal Component Analysis (TB 1: Section 6.3)	1hour
4.7	Dimensionality reduction – Principal Component Analysis (TB 1: Section 6.3)	1hour
4.8	Factor Analysis (TB 1: Section 6.4)	1hour
4.9	Multidimensional scaling (TB 1: Section 6.5)	1hour
4.10	Linear Discriminant Analysis (TB 1: Section 6.6)	1hour
Module 5 : Classification Assessment (8 hours)		
5.1	Performance measures - Precision, Recall, Accuracy, F-Measure, ROC, AUC. (TB 2: Chapter 22.1)	1hour
5.2	Boot strapping, Cross validation	1hour
5.3	Ensemble methods- bagging	1hour
5.4	Ensemble methods- boosting	1hour
5.5	Bias-Variance decomposition (TB 2: Chapter 22.3)	1hour
5.6	Bias-Variance decomposition (TB 2: Chapter 22.3)	1hour
5.7	Face detection (TB 3: Chapter 5 Section Application: A Face Detection Pipeline)	1hour
5.8	Face detection (TB 3: Chapter 5 Section Application: A Face Detection Pipeline)	1hour



CST 385	CLIENT SERVER SYSTEMS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

The syllabus is prepared with the view of preparing the Engineering Graduates to build effective Client/Server applications. This course aims at providing a foundation in decentralized computer systems, using the client/server model. The course content is decided to cover the essential fundamentals which can be taught within the given slots in the curriculum.

Prerequisite: **Basic knowledge in Computer**

Course Outcomes: After the completion of the course the student will be able to

Course Outcomes	
CO 1	Identify the basics of client/server systems and the driving force behind the development of client/server systems(Cognitive Knowledge Level: Understand)
CO 2	Outline the architecture and classifications of client/server systems(Cognitive Knowledge Level: Understand)
CO 3	Summarize the client/server network services for an application(Cognitive Knowledge Level: Understand)
CO 4	Identify management services and issues in network (Cognitive Knowledge Level: Understand)
CO 5	Outline the Client/Server technology in respect of databases and Client/Server database architecture (Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	☑	☑										☑
CO2	☑	☑										☑
CO3	☑	☑										☑
CO4	☑											☑
CO5	☑	☑										☑

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Percentage)	Test 2 (Percentage)	
Remember	40	40	40
Understand	40	40	40
Apply	20	20	20
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Test 1 (for theory, for 2 hrs) : 20 marks

Continuous Assessment Test 2 (for lab, internal examination, for 2hrs) : 20 marks

Internal Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 5 questions with 2 questions from each module (2.5 modules x 2 = 5), having 3 marks for each question. Students should answer all questions. Part B also contains 5 questions with 2 questions from each module (2.5 modules

x 2 = 5), of which a student should answer any one. The questions should not have sub-divisions and each one carries 7 marks.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Computing in client server architecture over Mainframe architecture has certain advantages and disadvantages. Describe at least three advantages and disadvantages for each architecture.

Course Outcome 2 (CO2):

1. Explain the role of mainframe-centric model in Client/Server computing?

Course Outcome 3(CO3):

1. Describe the client server system development methodology? Explain different phases of System Integration Life-Cycle.

Course Outcome 4 (CO4):

1. Explain about network management and remote system management. How can security be provided to the network?

Course Outcome 5 (CO5):

1. Explain various types of Client/Server Database Architecture

Syllabus

Module – 1 (Introduction)

Introduction to Client/Server computing - Basic Client/Server Computing Model, Server for Every Client- File Server, Print Server, Application Server, Mail Server, Directory Services Server, Web Server, Database Server, Transaction Servers. Client/Server-Fat or Thin, Stateless

or Stateful, Servers and Mainframes, Client/Server Functions. Driving Forces behind Client/Server Computing- Business Perspective, Technology Perspective.

Module -2 (Client/Server Classification)

Client/Server Types-Single Client/Single Server, Multiple Clients/Single Server, Multiple Clients/Multiple Servers, Integration With Distributed Computing, Alternatives To Client/Server Systems. Classification of Client/Server Systems- Two-Tier Computing, Middleware, Three-Tier Computing- Model View Controller (MVC), Principles behind Client/Server Systems. Client/Server Topologies. Existing Client/Server Architecture. Architecture for Business Information System.

Module -3 (Client/Server Application Components)

Client- Services, Request for services, RPC, Windows services, Print services, Remote boot services, other remote services, Utility Services. Server- Detailed server functionality, Network operating system, Available platforms, Server operating system. Organizational Expectations, Improving performance of client/server applications, Single system image, Downsizing and Rightsizing, Advantages and disadvantages of Client/Server computing, Applications of Client/Server.

Module -4 (Client/ Server Systems Services and Support)

Services and Support- System administration, Availability, Reliability, Scalability, Observability, Agility, Serviceability. Software Distribution, Performance, Network management. Remote Systems Management- RDP, Telnet, SSH, Security. LAN and Network Management issues.

Module -5(Client/Server Technology and Databases)

Client/Server Technology and Databases - Storing Data, Database System Architectures. Client/Server In Respect Of Databases- Client/Server Databases, Client/Server Database Computing, Database Computing Vs. Mainframe, PC/File Server Computing. Client/Server Database Architecture - Process-Per-Client Architecture, Multi-Threaded Architecture, Hybrid Architecture. Database Middleware Component - Application Programming Interface, Database Translator, Network Translator.

Text Book

1. Patrick Smith & Steve Guengerich, Client / Server Computing, PHI
2. Subhash Chandra Yadav, Sanjay Kumar Singh, An Introduction to Client/Server Computing, New Age International Publishers

Reference Books

1. Jeffrey D.Schank, “Novell’s Guide to Client-Server Application & Architecture” Novell Press
2. Robert Orfali, Dan Harkey, Jeri Edwards, Client/Server Survival Guide, Wiley-India Edition, Third Edition
3. Dawna Travis Dewire, Client Server Computing — McGraw Hill
4. W.H.Inman, Developing Client Server Applications, BPB

Model Question Paper

QP CODE: _____

PAGES: ____

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 385

Course Name : Client Server Systems

Max Marks: 100

Duration: 3 Hours

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Differentiate between Stateful and Stateless servers
2. List the different phases and activities of client/server system development methodology.
3. How does transmission protocol work in client/server applications?
4. List any six services in single system image environment.
5. Specify the role of the client in Client/Server computing and also list any six services provided by the client.
6. Why do most RPC system support call by value semantics for parameter passing?
7. What do you mean by a thin client network? List three advantages of the Thin

Client Network system.

8. How are connectivity and interoperability between .client/server achieved?
9. One disadvantage of the Client/Server system is lack of control in a Database Management environment. Justify.
10. Explain the DBMS concept in client/server architecture.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Differentiate between Transaction server and Data server system with examples. (7)
- (b) Computing in client server architecture over Mainframe architecture has certain advantages and disadvantages. Describe at least three advantages and disadvantages for each architecture. (7)

OR

12. (a) Explain various Clients/Server system development tools. (6)
- (b) Classify and describe the driving forces that drive the move to Client/Server computing. (8)
13. (a) Explain the role of mainframe-centric model in Client/Server computing? (5)
- (b) Describe the three types of Client/Server systems in existence (9)

OR

14. (a) List and explain the general forces behind the architecture for business information systems (7)
- (b) Explain the different distribution styles. (7)
15. (a) Illustrate the concept of rightsizing and downsizing in Client/Server Computing (7)
- (b) What is client server system development methodology? Explain the (7)

different phases of System Integration Life-Cycle.

OR

16. (a) In Client/Server computing, explain the following with examples **(10)**
- i. Dynamic Data Exchange
 - ii. RPC, Remote Procedure Call
 - iii. Remote Boot Service
 - iv. Diskless Computer
 - v. Object-linking and embedding
- (b) Explain the functions and features of Network Operating System **(4)**
17. (a) Explain about network management and remote system management. How can security be provided to the network? **(10)**
- (b) In client server architecture, what do you mean by Availability, Reliability, Serviceability and Security? Explain with examples. **(4)**

OR

18. (a) Client server is modular infrastructure, this is intended to improve Usability, Flexibility, Interoperability and Scalability. Explain each term with an example, in each case how it helps to improve the functionality of client server architecture. **(7)**
- (b) Explain about network management and remote system management. How can security be provided to network? **(7)**
19. (a) Explain the different types of Client/Server Database Architecture **(9)**
- (b) List and explain the main components of Database middleware **(5)**
- OR**
20. (a) Discuss types of database utilities, tools and their functions **(7)**
- (b) Discuss about the role of traditional and web databases in handling client/server based applications. **(7)**

Teaching Plan

Module- 1(Introduction)		(10 hours)
1.1	Basic Client/Server Computing Model	1 hour
1.2	Server for Every Client- File Server, Print Server	1 hour
1.3	Application Server, Mail Server, Directory Services Server	1 hour
1.4	Web Server, Database Server	1 hour
1.5	Transaction Servers	1 hour
1.6	Client/Server-Fat or Thin	1 hour
1.7	Stateless or Stateful	1 hour
1.8	Servers and Mainframes	1 hour
1.9	Client/Server Functions	1 hour
1.10	Driving Forces behind Client/Server Computing- Business Perspective, Technology Perspective	1 hour
Module- 2 (Client/Server Classification)		(10 hours)
2.1	Client/Server Types-Single Client/Single Server	1 hour
2.2	Multiple Clients/Single Server, Multiple Clients/Multiple Servers	1 hour
2.3	Integration With Distributed Computing	1 hour
2.4	Alternatives To Client/Server Systems	1 hour
2.5	Classification of Client/Server Systems- Two-Tier Computing, Middleware	1 hour
2.6	Three-Tier Computing- Model View Controller (MVC)	1 hour
2.7	Principles behind Client/Server Systems.	1 hour
2.8	Client/Server Topologies	1 hour
2.9	Existing Client/Server Architecture	1 hour
2.10	Architecture for Business Information System	1 hour
Module -3 (Client/Server Application Components)		(9 hours)
3.1	The client: Services, Request for services, RPC	1 hour
3.2	Windows services, Print services, Remote boot services	1 hour

3.3	Utility Services & Other Services	1 hour
3.4	Server- Detailed server functionality, Network operating system	1 hour
3.5	Available platforms, Server operating system	1 hour
3.6	Organizational Expectations, Improving performance of client/server applications	1 hour
3.7	Single system image, Downsizing and Rightsizing	1 hour
3.8	Advantages and disadvantages of Client/Server computing	1 hour
3.9	Applications of Client/Server	1 hour
Module -4 (Client/ Server Systems Services and Support)		(8 hours)
4.1	Services and Support, System administration	1 hour
4.2	Availability, Reliability	1 hour
4.3	Scalability, Observability, Agility	1 hour
4.4	Serviceability, Software Distribution	1 hour
4.5	Performance	1 hour
4.6	Network management	1 hour
4.7	Remote Systems Management- RDP, Telnet, SSH	1 hour
4.8	Security, LAN and Network Management issues	1 hour
Module -5(Client/Server Technology and Databases)		(8 hours)
5.1	Client/Server Technology and Databases - Storing Data	1 hour
5.2	Database System Architectures	1 hour
5.3	Client/Server In Respect Of Databases- Client/Server Databases	1 hour
5.4	Client/Server Database Computing	1 hour
5.5	Database Computing Vs. Mainframe, PC/File Server Computing	1 hour
5.	Client/Server Database Architecture - Process-Per-Client Architecture	1 hour
5.7	Multi-Threaded Architecture, Hybrid Architecture	1 hour
5.8	Database Middleware Component - Application Programming Interface, Database Translator, Network Translator	1 hour

APJ ABDUL KALAM
TECHNOLOGICAL
UNIVERSITY

SEMESTER V

HONOURS



CST 393	CRYPTOGRAPHIC ALGORITHMS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0		4

Preamble:

The course on Cryptographic Algorithms aims at exploring various algorithms deployed in offering confidentiality, integrity, authentication and non-repudiation services. This course covers classical encryption techniques, symmetric and public key crypto-system, key exchange and management, and authentication functions. The concepts covered in this course enable the learners in effective use of cryptographic algorithms for real life applications.

Prerequisite: A sound background in Number Theory.

Course Outcomes: After the completion of the course the student will be able to

CO1	Identify the security services provided for different types of security attacks. (Cognitive Knowledge Level : Understand)
CO2	Summarize the classical encryption techniques for information hiding. (Cognitive Knowledge Level: Apply)
CO3	Illustrate symmetric / asymmetric key cryptographic algorithms for secure communication.(Cognitive Knowledge Level: Apply)
CO4	Interpret key management techniques for secure communication.(Cognitive Knowledge Level: Understand)
CO5	Summarize message authentication functions in a secure communication scenario.(Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓

CO2	✓	✓	✓		✓	✓						✓
CO3	✓	✓	✓	✓	✓	✓						✓
CO4	✓	✓	✓	✓	✓	✓						✓
CO5	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percent	

		age)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Tests : **25 marks**

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks.

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Introduction to the Concepts of Security)

Need for security, Security approaches, Principles of security, Types of attacks, OSI Security Architecture, Classical encryption techniques - Substitution techniques, Transposition techniques. Stream cipher, Block cipher, Public key cryptosystems vs. Symmetric key cryptosystems, Encrypting communication channels.

Module-2 (Symmetric Key Cryptosystems)

Overview of symmetric key cryptography, Block cipher principles, Data Encryption Standard (DES), Differential and Linear cryptanalysis, Double DES, Triple DES, International Data Encryption Algorithm (IDEA), Advanced Encryption Algorithm (AES), Block cipher modes of operation, Stream cipher, RC4.

Module-3 (Public Key Cryptosystems)

Principles of public key cryptosystems, RSA algorithm, RSA illustration, Attacks, ElGamal cryptographic system, Knapsack algorithm, Diffie-Hellman key exchange algorithm, Elliptical curve cryptosystems.

Module-4 (Key Management)

Symmetric key distribution using symmetric encryption, Symmetric key distribution using asymmetric encryption, Distribution of public keys, Generating keys, transferring keys, Verifying keys, Updating keys, Storing keys, Backup keys, Compromised keys, Public key infrastructure.

Module – 5 (Authentication)

Authentication requirements, Authentication functions, Message authentication codes (MAC), Hash functions, Security of Hash functions and MAC, Message Digest 5 (MD5), Secure Hash Algorithm (SHA)-512, Hash-based Message Authentication Code (HMAC), Cipher-based Message Authentication Code (CMAC), X.509 Authentication services.

Text Books

1. William Stallings, Cryptography and Network Security Principles and Practice, Pearson Edu, 6e.
2. Bruce Schneier, Applied Cryptography Protocols, Algorithms and source code in C, Wiley, 2e.

References

1. Behrouz A. Forouzan, Cryptography and Network Security, McGraw Hill, 2e.
2. Johannes A. Buchmann, Introduction to Cryptography, Springer, 2e.
3. Douglas R. Stinson, Cryptography Theory and Practice, 3e, Chapman & Hall/CRC, 2006.
4. Bernard Menezes, Network Security and Cryptography, Cengage Learning, 2011.

Sample Course Level Assessment Questions**Course Outcome 1 (CO1):**

1. Consider an automated teller machine (ATM) in which users provide a personal identification number (PIN) and a card for account access. Give examples of confidentiality, integrity, and availability requirements associated with the system and, in each case, indicate the degree of importance of the requirement.
2. Discuss the different security services provided for preventing security attacks.

Course Outcome 2 (CO2):

1. The encryption key in a transposition cipher is (3,2,6,1,5,4). Find the decryption key
2. Discuss the process of encryption in Vernam cipher

Course Outcome 3 (CO3):

1. Devise a meet-in-the-middle attack for a triple DES.

2. Write an algorithm for the InvSubBytes transformation and implement using python **(Assignment)**
3. Consider the following elliptic curve signature scheme. We have a global elliptic curve, prime p , and “generator” G . Alice picks a private signing key X_A and forms the public verifying $Y_A = X_A G$. To sign a message M :
 - Alice picks a value k
 - Alice sends Bob M , k and the signature $S = M - kX_A G$.
 - Bob verifies that $M = S + kY_A$.

Show that the verification process produces an equality if the signature is valid.

4. Write an algorithm to add two points on an elliptic curve over $GF(p)$ and implement using Python. **(Assignment)**
5. Write an algorithm for encryption using knapsack cryptosystem and implement using Java. **(Assignment)**

Course Outcome4 (CO4):

1. List four general categories of schemes for the distribution of public keys.
2. What are the essential ingredients of a public-key directory?

Course Outcome 5 (CO5):

1. State the value of the length field in SHA-512 if the length of the message is 1919 bits and 1920 bits.
2. Write an algorithm in pseudo code for HMAC and implement using Python **(Assignment)**

Model Question Paper

QP CODE:

Reg No: _____

Name : _____

PAGES : 3

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

**FIFTH SEMESTER B.TECH DEGREE EXAMINATION(HONORS), MONTH &
YEAR**

Course Code: CST 393

Course Name: Cryptographic Algorithms

Max.Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. State the two approaches in attacking a cipher.
2. Define Substitution Cipher. Encrypt using one time pad $M = \text{HONORS}$ and $K = \text{CIPHER}$.
3. Specify the purpose of S-Boxes in Data Encryption Standard (DES).
4. Differentiate between diffusion and confusion.
5. Perform encryption using RSA Algorithm for the following $p=7$; $q=11$; $e=13$; $M=5$.
6. Is Diffie-Hellman key exchange protocol vulnerable? Justify.
7. List the techniques for distribution of public keys.
8. Define a certificate authority and its relation to public key cryptography.
9. Distinguish between integrity and message authentication.
10. What types of attacks are addressed by message authentication?

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) With a neat sketch, Explain OSI Security architecture model. (8)
- (b) How does link encryption differ from end-to-end encryption? Explain. (6)
- OR**
12. (a) Encrypt the text “cryptography” using the Hill Cipher with the key $\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}$. Show the calculations. (8)
- (b) Illustrate the steps involved in encrypting a plain text using playfair cipher with an example. (6)
13. (a) With a neat sketch, explain a single round in DES. 10
- (b) Explain encryption and decryption using 2 keys and 3 keys of triple DES. (4)
- OR**
14. (a) Explain the block cipher modes i) Cipher feedback mode ii) Output feedback mode. (8)
- (b) Describe the four types of transformations in AES. (6)
15. (a) Write an algorithm for generating public and private key using Elliptical curve cryptography. (10)

- (b) The equation $y^2 = x^3 + x + 1$, the calculation is done modulo 13. Add two points $R = P + Q$, where $P = (4, 2)$ and $Q = (10, 6)$. (4)

OR

16. User A and B use the Diffie-Hellman key exchange technique with a common prime $q = 71$ and primitive root $\alpha = 7$.
- (a) If user A has private key $X_A = 3$, What is A's public key Y_A ? (7)
- (b) If user B has private key $X_B = 6$, What is A's public key Y_B ? (7)
17. (a) Define a session key and show how a KDC can create a session key between Alice and Bob. (7)
- (b) What are the requirements for the use of a public-key certificate scheme? (7)

OR

18. (a) What are the core components of a PKI? Briefly describe each component. (8)
- (b) Describe the following (i) Updating keys (ii) Compromised Keys. (6)
19. (a) Describe how SHA-512 logic produce message digest (10)
- (b) Distinguish between HMAC and CMAC (4)

OR

20. (a) Specify the format for X.509 certificate. Explain the steps required to obtain user's certificate. (7)
- (b) With suitable block diagrams, explain the types of functions that may be used to produce an authenticator. (8)

Teaching Plan

No	Contents	No of Lecture Hrs
Module - 1 (Introduction to the Concepts of Security) (9 hrs)		
1.1	Need for security, Security approaches	1 hour
1.2	Principles of security, Types of attacks	1 hour
1.3	OSI Security Architecture	1 hour
1.4	Classical encryption techniques: Substitution techniques(Caesar cipher, Monoalphabetic cipher, Playfair cipher)	1 hour
1.5	Classical encryption techniques: Substitution techniques (Hill cipher, Polyalphabetic cipher, One-time pad)	1 hour
1.6	Classical encryption techniques: Transposition techniques	1 hour
1.7	Stream cipher, Block cipher	1 hour
1.8	Public- key cryptosystems vs. Symmetric key cryptosystems	1 hour
1.9	Encrypting communication channels	1 hour
Module - 2 (Symmetric key cryptosystems) (11 hrs)		
2.1	Overview of symmetric key cryptography	1 hour
2.2	Block cipher principles	1 hour
2.3	Data Encryption Standard (DES)	1 hour
2.4	DES design criteria	1 hour
2.5	Differential and Linear cryptanalysis	1 hour
2.6	Double DES, Triple DES	1 hour

2.7	IDEA	1 hour
2.8	Advanced Encryption Algorithm (AES structure)	1 hour
2.9	Advanced Encryption Algorithm (Transformations)	1 hour
2.10	Block cipher modes of operation	1 hour
2.11	Stream cipher, RC4	1 hour
Module - 3 (Public key cryptosystems) (8 hrs)		
3.1	Principles of public key cryptosystems	1 hour
3.2	RSA algorithm	1 hour
3.3	RSA illustration, Attacks	1 hour
3.4	ElGamal cryptographic system	1 hour
3.5	Knapsack algorithm	1 hour
3.6	Diffie-Hellman key exchange algorithm	1 hour
3.7	Elliptical curve cryptosystems(Elliptical curve arithmetic)	1 hour
3.8	Elliptical curve cryptosystems (Elliptical curve algorithm)	1 hour
Module - 4 (Key Management) (8 hrs) [Text book-2]		
4.1	Symmetric key distribution using symmetric encryption	1 hour
4.2	Symmetric key distribution using asymmetric encryption	1 hour
4.3	Distribution of public keys	1 hour
4.4	Generating keys, Transferring keys	1 hour

4.5	Verifying keys, Updating keys	1 hour
4.6	Storing keys, Backup keys	1 hour
4.7	Compromised keys	1 hour
4.8	Public key infrastructure	1 hour
Module - 5 (Authentication) (9 hrs)		
5.1	Authentication requirements	1 hour
5.2	Authentication functions	1 hour
5.3	Message Authentication Codes (MAC)	1 hour
5.4	Hash functions	1 hour
5.5	Security of Hash functions and MAC	1 hour
5.6	MD5	1 hour
5.7	SHA-512	1 hour
5.8	HMAC, CMAC	1 hour
5.9	X.509 Authentication services	1 hour

CST 395	NEURAL NETWORKS AND DEEP LEARNING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0		

Preamble:

Neural networks is a biologically inspired programming paradigm which enables a computer to learn from observational data and deep learning is a powerful set of techniques for training neural networks. This course introduces the key concepts in neural networks, its architecture and learning paradigms, optimization techniques, basic concepts in deep learning, Convolutional Neural Networks and Recurrent Neural Networks. The students will be able to provide best solutions to real world problems in domains such as computer vision and natural language processing.

Prerequisite: A Sound knowledge in Computational fundamentals of machine learning

Course Outcomes: After the completion of the course the student will be able to

CO1	Demonstrate the basic concepts of machine learning models and performance measures. (Cognitive Knowledge Level : Understand)
CO2	Illustrate the basic concepts of neural networks and its practical issues (Cognitive Knowledge Level : Apply)
CO3	Outline the standard regularization and optimization techniques for deep neural networks (Cognitive Knowledge Level : Understand)
CO4	Build CNN and RNN models for different use cases. (Cognitive Knowledge Level : Apply)
CO5	Explain the concepts of modern RNNs like LSTM, GRU (Cognitive Knowledge Level : Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓	✓	✓						✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (%)	Test2 (%)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern

Attendance	10 marks
Continuous Assessment Tests	25 marks
Continuous Assessment Assignment	15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B

contains 2 questions from each module of which a student should answer any one. Each question can have a maximum 2 subdivisions and carry 14 marks.

Syllabus

Module - 1 (Basics of Machine Learning)

Machine Learning basics - Learning algorithms - Supervised, Unsupervised, Reinforcement, Overfitting, Underfitting, Hyper parameters and Validation sets, Estimators -Bias and Variance. Challenges in machine learning. Simple Linear Regression, Logistic Regression, Performance measures - Confusion matrix, Accuracy, Precision, Recall, Sensitivity, Specificity, Receiver Operating Characteristic curve(ROC), Area Under Curve(AUC).

Module -2 (Neural Networks)

Introduction to neural networks -Single layer perceptrons, Multi Layer Perceptrons (MLPs), Representation Power of MLPs, Activation functions - Sigmoid, Tanh, ReLU, Softmax. Risk minimization, Loss function, Training MLPs with backpropagation, Practical issues in neural network training - The Problem of Overfitting, Vanishing and exploding gradient problems, Difficulties in convergence, Local and spurious Optima, Computational Challenges. Applications of neural networks.

Module 3 (Deep learning)

Introduction to deep learning, Deep feed forward network, Training deep models, Optimization techniques - Gradient Descent (GD), GD with momentum, Nesterov accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam. Regularization Techniques - L1 and L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout, Parameter initialization.

Module -4 (Convolutional Neural Network)

Convolutional Neural Networks – Convolution operation, Motivation, Pooling, Convolution and Pooling as an infinitely strong prior, Variants of convolution functions, Structured outputs, Data types, Efficient convolution algorithms. Practical use cases for CNNs, Case study - Building CNN model AlexNet with handwritten digit dataset MNIST.

Module- 5 (Recurrent Neural Network)

Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs. Case study - Natural Language Processing.

Text Book

1. Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press, 2016.
2. Neural Networks and Deep Learning, Aggarwal, Charu C., c Springer International Publishing AG, part of Springer Nature 2018
3. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms (1st. ed.). Nikhil Buduma and Nicholas Locascio. 2017. O'Reilly Media, Inc.

Reference Books

1. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.
2. Yegnalarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Michael Nielsen, Neural Networks and Deep Learning, 2018

Course Level Assessment Questions**Course Outcome 1 (CO1):**

1. Predict the price of a 1000 square feet house using the regression model generated from the following data.

No.	Square feet	Price(Lakhs)
1	500	5
2	900	10
3	1200	13
4	1500	18
5	2000	25
6	2500	32
7	2700	35

2. Consider a two-class classification problem of predicting whether a photograph contains a man or a woman. Suppose we have a test dataset of 10 records with expected outcomes and a set of predictions from our classification algorithm. Compute the confusion matrix, accuracy, precision, recall, sensitivity and specificity on the following data.

Sl.No.	Actual	Predicted
1	man	woman
2	man	man
3	woman	woman
4	man	man

5	man	woman
6	woman	woman
7	woman	man
8	man	man
9	man	woman
10	woman	woman

Course Outcome 2 (CO2):

1. Suppose you have a 3-dimensional input $x = (x_1, x_2, x_3) = (2, 2, 1)$ fully connected with weights $(0.5, 0.3, 0.2)$ to one neuron which is in the hidden layer with sigmoid activation function. Calculate the output of the hidden layer neuron.
2. Consider the case of the XOR function in which the two points $\{(0, 0), (1, 1)\}$ belong to one class, and the other two points $\{(1, 0), (0, 1)\}$ belong to the other class. Design a multilayer perceptron for this binary classification problem.

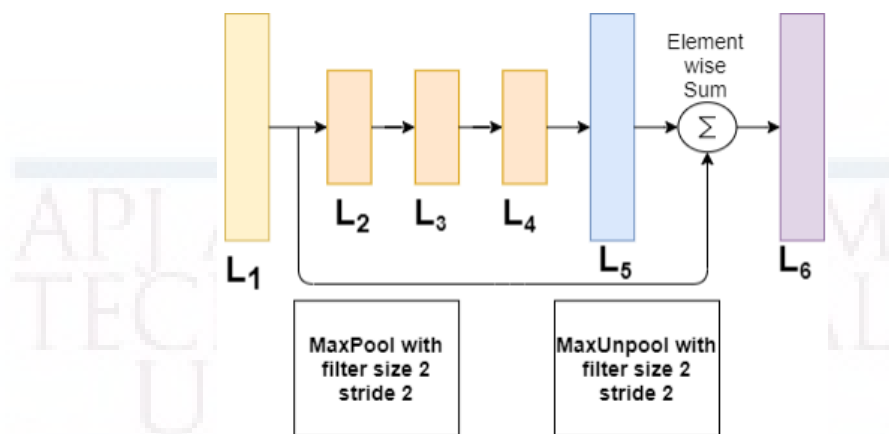
Course Outcome 3 (CO3):

1. Derive a mathematical expression to show L2 regularization as weight decay.
2. Explain how L2 regularization improves the performance of deep feed forward neural networks.
3. Explain how L1 regularization method leads to weight sparsity.

Course Outcome 4 (CO4):

1. Draw and explain the architecture of convolutional neural networks.
2. You are given a classification problem to classify the handwritten digits. Suggest a learning and/or inference machine with its architecture, an objective function, and an optimization routine, along with how input and output will be prepared for the classifier.
3. In a Deep CNN architecture the feature map L_1 was processed by the following operations as shown in the figure. First down sampled using max pool operation of size 2 and stride 2, and three convolution operations and finally max unpool operation and followed by an element wise sum. The feature map L_1 and L_4 are given below. Compute the matrix L_6 .

$$L_1 = \begin{matrix} 10 & 20 & 15 & 22 \\ 20 & 16 & 28 & 30 \\ 30 & 12 & 20 & 16 \\ 20 & 20 & 40 & 12 \end{matrix} \quad L_4 = \begin{matrix} 10 & 20 \\ 20 & 30 \end{matrix}$$



4. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.

Course Outcome 5 (CO5):

1. Draw and explain the architecture of LSTM.
2. List the differences between LSTM and GRU

Model Question Paper

QP CODE:

PAGES:4

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION(HONORS), MONTH & YEAR

Course Code: CST 395

Course Name: Neural Networks and Deep Learning

Max.Marks:100

Duration:3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. List and compare the types of machine learning algorithms
2. Suppose 10000 patients get tested for flu; out of them, 9000 are actually healthy and 1000 are actually sick. For the sick people, a test was positive for 620 and negative for 380. For healthy people, the same test was positive for 180 and negative for 8820. Construct a confusion matrix for the data and compute the

accuracy, precision and recall for the data

3. Illustrate the limitation of a single layer perceptron with an example
4. Specify the advantages of ReLU over sigmoid activation function.
5. Derive weight updating rule in gradient descent when the error function is a) mean squared error b) cross entropy
6. List any three methods to prevent overfitting in neural networks
7. What happens if the stride of the convolutional layer increases? What can be the maximum stride? Justify your answer.
8. Consider an activation volume of size $13 \times 13 \times 64$ and a filter of size $3 \times 3 \times 64$. Discuss whether it is possible to perform convolutions with strides 2, 3 and 5. Justify your answer in each case.
9. How does a recursive neural network work?
10. List down three differences between LSTM and RNN

(10x3=30
)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Prove that the decision boundary of binary logistic regression is linear
- (b) Given the following data, construct the ROC curve of the data. Compute the AUC.

(9)

Threshold	TP	TN	FP	FN
1	0	25	0	29
2	7	25	0	22
3	18	24	1	11
4	26	20	5	3
5	29	11	14	0

(5)

6	29	0	25	0
7	29	0	25	0

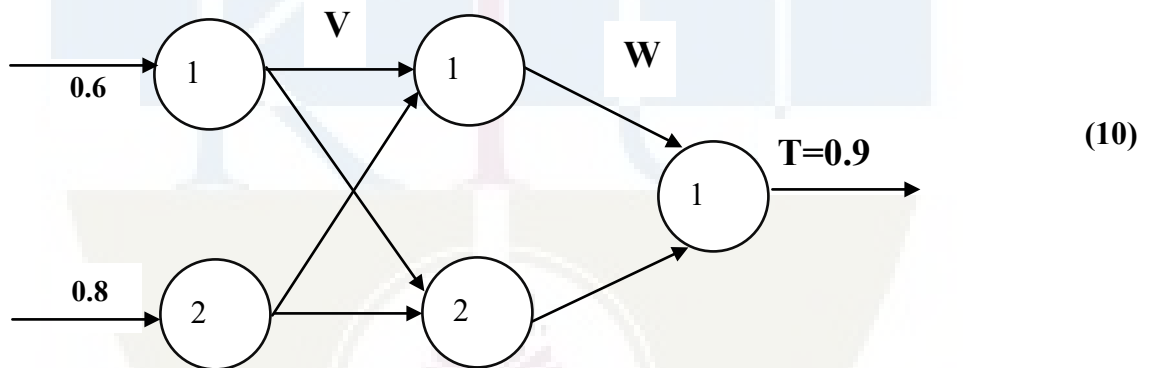
OR

12. (a) With an example classification problem, explain the following terms:
 a) Hyper parameters b) Training set c) Validation sets d) Bias e) Variance (8)

- (b) Determine the regression equation by finding the regression slope coefficient and the intercept value using the following data. (6)

x	55	60	65	70	80
y	52	54	56	58	62

13. (a) Update the parameters V_{11} in the given MLP using back propagation with learning rate as 0.5 and activation function as sigmoid. Initial weights are given as $V_{11}= 0.2$, $V_{12}=0.1$, $V_{21}=0.1$, $V_{22}=0.3$, $V_{11}=0.2$, $W_{11}=0.5$, $W_{21}=0.2$

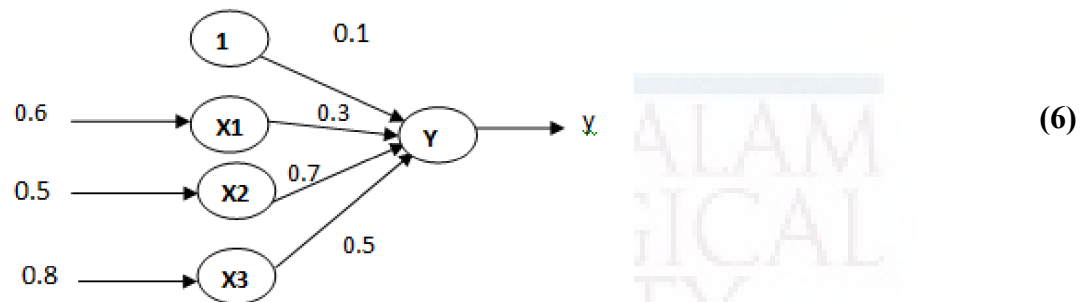


- (b) Explain the importance of choosing the right step size in neural networks (4)

OR

14. (a) Explain in detail any four practical issues in neural network training (8)

- (b) Calculate the output of the following neuron Y with the activation function as a) binary sigmoid b) tanh c)ReLU



15. (a) Explain, what might happen in ADAGRAD, where momentum is expressed as $\Delta \theta_j = -\eta \theta_j / \sqrt{\sum_{i=1}^n \theta_j^2}$ where the denominator computes the L2 norm of all previous gradients on a per-dimension basis and η is a global learning rate shared by all dimensions. (6)

- (b) Differentiate gradient descent with and without momentum. Give equations for weight updation in GD with and without momentum. Illustrate plateaus, saddle points and slowly varying gradients. (8)

OR

16. (a) Suppose a supervised learning problem is given to model a deep feed forward neural network. Suggest solutions for the following a) small sized dataset for training b) dataset with both labelled and unlabeled data c) large data set but data from different distribution (9)

- (b) Describe the effect in bias and variance when a neural network is modified with more number of hidden units followed with dropout regularization. (5)

17. (a) Draw and explain the architecture of Convolutional Neural Networks (8)

- (b) Suppose that a CNN was trained to classify images into different categories. It performed well on a validation set that was taken from the same source as the training set but not on a testing set. What could be the problem with the training of such a CNN? How will you ascertain the problem? How can those problems be solved? (6)

OR

18. (a) Explain the following convolution functions a) tensors b) kernel flipping c) down sampling d) strides e) zero padding. (10)

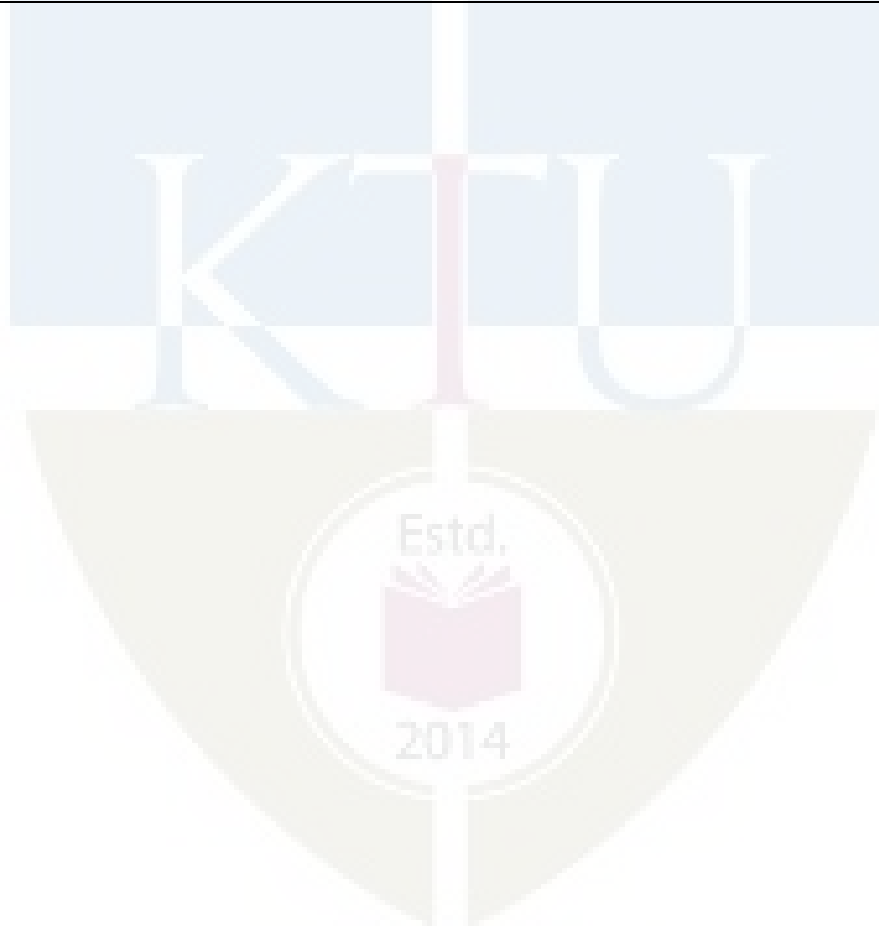
- (b) What is the motivation behind convolution neural networks? (4)
19. (a) Describe how an LSTM takes care of the vanishing gradient problem. Use some hypothetical numbers for input and output signals to explain the concept (8)
- (b) Explain the architecture of Recurrent Neural Networks (6)
- OR**
20. (a) Explain LSTM based solution for anyone of the problems in the Natural Language Processing domain. (8)
- (b) Discuss the architecture of GRU (6)

Teaching Plan

Module 1 : [Text book 1: Chapter 5, Textbook 2: Chapter 2](9 hours)		
1.1	Introduction, Learning algorithms - Supervised, Unsupervised, Reinforcement	1 hour
1.2	Overfitting, Underfitting, Hyperparameters	1 hour
1.3	Validation sets, Estimators -Bias and Variance. Challenges in machine learning.	1 hour
1.4	Simple Linear Regression	1 hour
1.5	Illustration of Linear Regression	1 hour
1.6	Logistic Regression	1 hour
1.7	Illustration of Logistic Regression	1 hour
1.8	Performance measures - Confusion matrix, Accuracy, Precision, Recall, Sensitivity, Specificity, ROC, AUC.	1 hour
1.9	Illustrative Examples for performance measures	1 hour
Module 2 : Text book 2, Chapter 1 (8 hours)		
2.1	Introduction to neural networks -Single layer perceptrons	1 hour
2.2	Multi Layer Perceptrons (MLPs), Representation Power of MLPs	1 hour
2.3	Activation functions - Sigmoid, Tanh, ReLU, Softmax. Risk minimization, Loss function	1 hour

2.4	Training MLPs with backpropagation	1 hour
2.5	Illustration of back propagation algorithm	1 hour
2.6	Practical issues in neural network training - The Problem of Overfitting, Vanishing and exploding gradient problems	1 hour
2.7	Difficulties in convergence, Local and spurious Optima, Computational Challenges.	1 hour
2.8	Applications of neural networks	1 hour
Module 3 : Text book 1: Chapter 7, 8, Text book 2, Chapter 3, 4 (10 hours)		
3.1	Introduction to deep learning, Deep feed forward network	1 hour
3.2	Training deep models - Introduction, setup and initialization issues	1 hour
3.3	Solving vanishing and exploding gradient problems	1 hour
3.4	Concepts of optimization, Gradient Descent (GD), GD with momentum.	1 hour
3.5	Nesterov accelerated GD, Stochastic GD.	1 hour
3.6	AdaGrad, RMSProp, Adam.	1 hour
3.7	Concepts of Regularization, L1 and L2 regularization.	1 hour
3.8	Early stopping, Dataset augmentation	1 hour
3.9	Parameter sharing and tying, Injecting noise at input, Ensemble methods	1 hour
3.10	Dropout, Parameter initialization.	1 hour
Module 4 : Text book 1, Chapter 9, Text book 2: Chapter 8 (8 hours)		
4.1	Convolutional Neural Networks, architecture	1 hour
4.2	Convolution and Pooling operation with example	1 hour
4.3	Convolution and Pooling as an infinitely strong prior	1 hour
4.4	Variants of convolution functions, structured outputs, data types	1 hour
4.5	Efficient convolution algorithms.	1 hour
4.6	Practical use cases for CNNs	1 hour
4.7	Case study - Building CNN with MNIST and AlexNet.	1 hour
4.8	Case study - Building CNN with MNIST and AlexNet	1 hour
Module 5 : Text book 1 :Chapter 10, 11, Text book 2:Chapter 7 (10 hours)		

5.1	Recurrent neural networks – Computational graphs, RNN design	1 hour
5.2	Encoder – decoder sequence to sequence architectures	1 hour
5.3	Deep recurrent networks- Architecture	1 hour
5.4	Recursive neural networks	1 hour
5.5	Modern RNNs - LSTM	1 hour
5.6	Modern RNNs - LSTM	1 hour
5.7	GRU	1 hour
5.8	Practical use cases for RNNs.	1 hour
5.9	Case study - Natural Language Processing.	1 hour
5.10	Case study - Natural Language Processing.	1 hour



CST 397	PRINCIPLES OF MODEL CHECKING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This course covers the basic theory and algorithm for an automatic verification process namely, model checking. Model checking is a formal process for proving the correctness of a hardware/software which can be modelled as a finite-state transition system. This course introduces the topics - finite-state modelling of hardware/software, linear-time properties, classification of linear-time properties, Linear Temporal Logic (LTL), a formal language for property specification, LTL model checking algorithm and model checking case studies. Proving correctness of a hardware/software is essential in safety critical systems in domains such as avionics, health care and automotive.

Prerequisite: Nil

Course Outcomes: After the completion of the course, the student will be able to

CO#	Course Outcomes
CO1	Illustrate an application for model checking. (Cognitive Knowledge Level: Understand)
CO2	Describe finite-state modelling of hardware and software. (Cognitive Knowledge Level: Understand)
CO3	Identify the linear-time properties required to represent the requirements of a system. (Cognitive Knowledge Level: Apply)
CO4	Specify a given linear-time property in Linear Temporal Logic (LTL). (Cognitive Knowledge Level: Apply)
CO5	Perform LTL model checking with the tool SAL (Symbolic Analysis Laboratory). (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓	✓	✓						✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓
CO6	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (Marks in percentage)	Test 2 (Marks in percentage)	End Semester Examination (Marks in percentage)
Remember	30	30	30
Understand	40	40	40
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**
 Continuous Assessment Test : **25 marks**
 Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus. The second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer anyone. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus**Module 1 (Introduction to Model Checking)**

System Verification– Hardware and Software Verification, Model Checking, Characteristics of Model Checking. Transition Systems – Transition System, Direct Predecessors and Successors, Terminal State, Deterministic Transition System.

Executions - Execution Fragment, Maximal and Initial Execution Fragment, Execution, Reachable States. Modeling Hardware and Software Systems- Sequential Hardware Circuits, data Dependent Systems.

Module - 2 (Linear Time Properties)

Linear-Time (LT) Properties - Deadlock. Linear-Time Behavior - Paths and State Graph, Path Fragment, Maximal and Initial Path Fragment, Path. Traces - Trace and Trace Fragment, LT Properties - LT Property, Satisfaction Relation for LT Properties, Trace Equivalence and LT Properties. Safety Properties and Invariants - Invariants, Safety Properties, Trace Equivalence and Safety properties. Liveness Properties - Liveness Property, Safety vs. Liveness Properties. Fairness - Fairness, Unconditional, Weak and Strong Fairness, Fairness Strategies, Fairness and Safety. (Definition and examples only for all topics - no proof required).

Module - 3 (Regular Properties)

Regular Properties - Model Checking Regular Safety properties - Regular Safety property, Verifying Regular Safety Properties. Automata on Infinite Words - ω -Regular Languages and Properties, Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA), Generalised Buchi Automata (Definitions only). Model Checking ω -Regular Properties - Persistence Properties and Product, Nested Depth-First Search (Only algorithms required).

Module - 4 (Linear Time Logic)

Linear Temporal Logic (LTL) - Syntax, Semantics, Equivalence of LTL Formulae, Weak Until, Release and Positive Normal Form, Fairness, Safety and Liveness in LTL (Definitions only). Automata Based LTL Model Checking (Algorithms and examples only).

Module - 5 (Model Checking in SAL)

Introduction - Introduction to the tool Symbolic Analysis Laboratory (SAL). The Language of SAL - The expression language, The transition Language, The module language, SAL Contexts. SAL Examples - Mutual Exclusion, Peterson's Protocol, Synchronous Bus Arbiter, Bounded Bakery protocol, Bakery Protocol, Simpson's Protocol, Stack.

Text Books

1. Christel Baier and Joost-Pieter Katoen, Principles of Model Checking, The MIT Press. (Modules 1 - 4)
2. Leonardo de Moura, Sam Owre and N. Shankar, The SAL Language Manual, SRI International (<http://sal.csl.sri.com/doc/language-report.pdf>, Chapters 1, 3, 4, 5, 6, 7) (Module 5)

Reference Materials

1. SAL Examples (<http://sal.csl.sri.com/examples.shtml>) (Module 5)

Course Level Assessment Questions**Course Outcome 1 (CO1):**

1. Explain how model checking can be effective in developing a nuclear power plant.

Course Outcome 2 (CO2):

1. Consider a message delivery system. The sender s is trying to send a series of messages to the receiver r in such a way that the $(i+1)^{st}$ message is sent only after the i^{th} message is delivered. There is a possibility of error in sending a message and in that case, s keeps on trying until it is able to send the message. Express this process as a transition system.

Course Outcome 3 (CO3):

1. Consider a shared memory segment s protected using a mutex lock variable m . Two processes p_1 and p_2 are trying to access s . Find the Linear Time properties of the system which will ensure safety, liveness and fairness.

Course Outcome 4 (CO4):

1. Express the Linear Time properties found in the above system using LTL.

Course Outcome 5 (CO5):

1. Model the above system using SAL and verify that the system avoids deadlock under all conditions.
- 2.

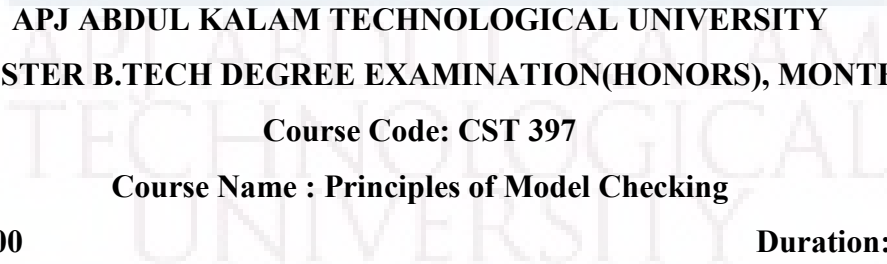
Model Question Paper

QP CODE: _____

PAGES: ____

Reg No: _____

Name: _____



APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FIFTH SEMESTER B.TECH DEGREE EXAMINATION(HONORS), MONTH & YEAR

Course Code: CST 397

Course Name : Principles of Model Checking

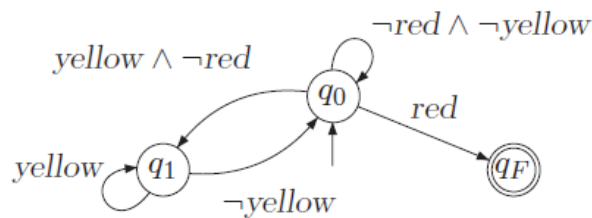
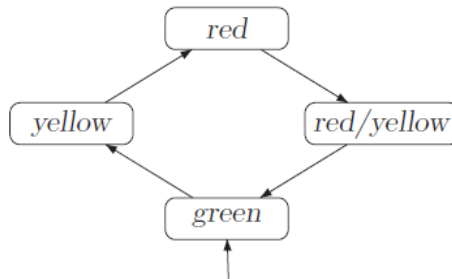
Max Marks: 100

Duration: 3 Hours

PART-A

(Answer All Questions. Each question carries 3 marks)

1. What is model checking? Give the schematic view of the model checking approach.
2. Give the transition system of a beverage vending machine.
3. What is an invariant in Linear Time (LT) properties? Give an example.
4. Give 3 Liveness properties in the Mutual Exclusion problem of processes.
5. Find the product automaton for the following Transition System and Non-Deterministic Finite Automaton (NFA).



6. Differentiate between Deterministic Buchi Automaton and Non-deterministic

Buchi Automaton. Give examples of each.

7. Express the following statements about traffic lights in Linear Temporal Logic (LTL).
 - a. Once red, the light can not become green immediately.
 - b. Once red, the light always becomes green eventually after being yellow for some time.
8. What is Positive Normal Form (PNF) in LTL? Give an example.
9. Write notes on Symbolic Analysis Laboratory (SAL).
10. What is a SAL context? Give an example.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain in detail the various phases of the model checking process. (7)
 - (b) Explain the strengths and weaknesses of model checking. (7)
- OR**
12. (a) Explain the following terms in association with execution of a transition system. (14)
 - a. Execution Fragment
 - b. Maximal and Initial Execution Fragment
 - c. Execution
 - d. Reachable States
 13. (a) With an example, explain the satisfaction relation for LT properties. (7)
 - (b) What is trace equivalence in Transition Systems? Give an example to show that if two transition systems have the same trace equivalence, they satisfy the same LT properties. (7)

14. (a) Give the transition system for the fault tolerant variant of the dining philosophers problem. (4)

(b) With a suitable example, explain the algorithms to check whether a Transition System satisfies an invariant or not. (10)

15. (a) Give the algorithm for verifying Regular Safety Properties. Explain with an appropriate example. (7)

(b) With a suitable example, explain Regular Safety Properties. (7)

OR

16. (a) Explain ω -Regular Properties. (4)

(b) Illustrate how ω -Regular Properties are verified. (10)

17. (a) Explain the syntax of Linear Temporal Logic (LTL). (7)

(b) Explain the semantics of LTL. (7)

OR

18. (a) With an example, give the difference between *until* and *weak until* in LTL. (4)

(b) With a suitable example, explain automata based LTL model checking. (10)

19. (a) Explain Peterson's protocol. What are the LTL properties to be verified to ensure its correctness? (8)

(b) Write a SAL script for the verification of Peterson's protocol. (6)

OR

20. (a) Show the SAL model corresponding to Bakery protocol. (8)

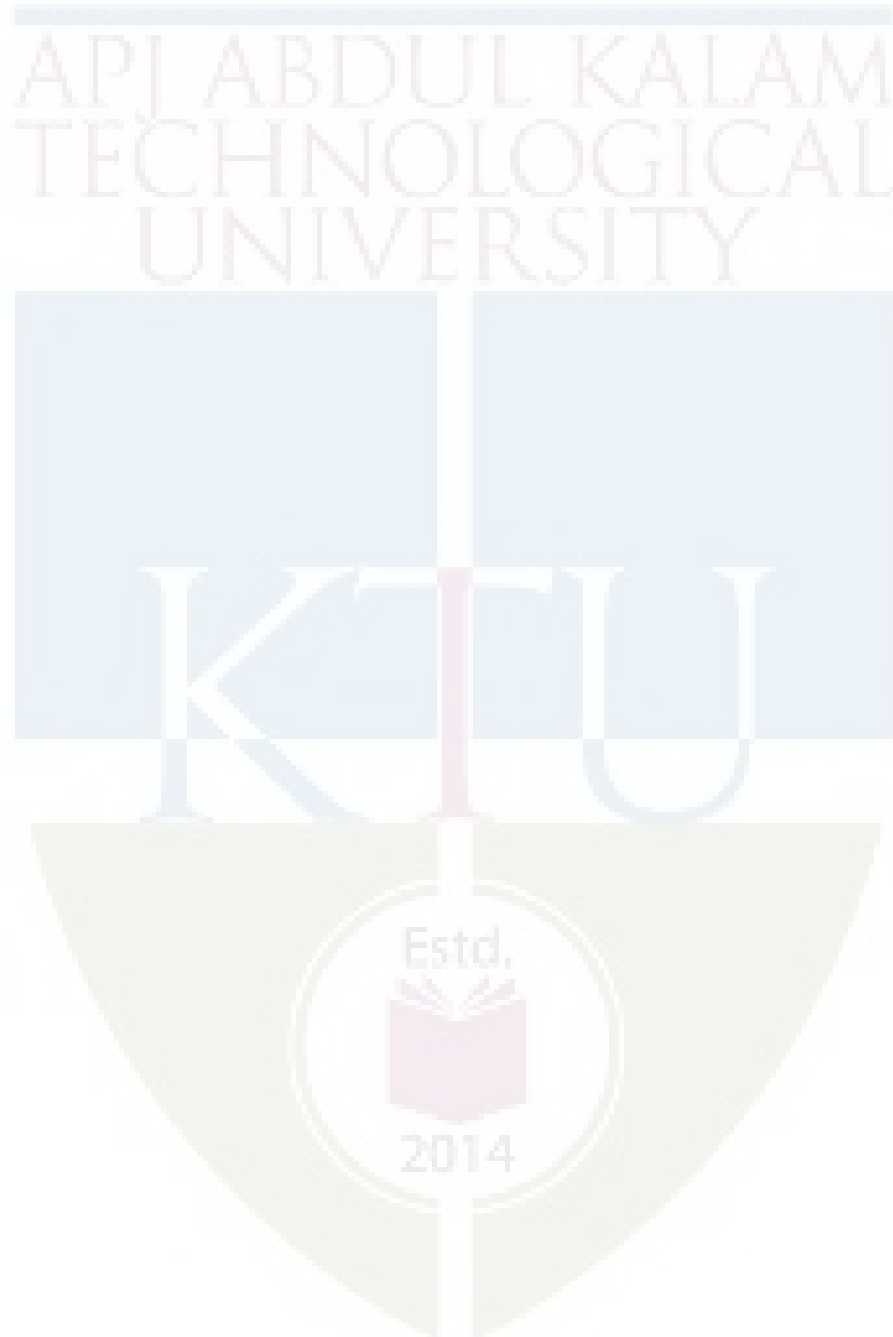
(b) List any three Linear Time properties of this model and show their LTL specifications. (6)

Teaching plan

Module 1 (Introduction to Model Checking)		8 Hours
1.1	System Verification – Hard- and Software Verification, Model Checking, Characteristics of Model Checking	1 Hour
1.2	Transition Systems – Transition System, Direct Predecessors and Successors,	1 Hour
1.3	Terminal State, Deterministic Transition System,	1 Hour
1.4	Executions - Execution Fragment, Maximal and Initial Execution Fragment	1 Hour
1.5	Execution, Reachable States	1 Hour
1.6	Modeling Hardware and Software Systems - Sequential Hardware Circuits	1 Hours
1.7	Data Dependent Systems (Lecture 1)	1 Hour
1.8	Data Dependent Systems (Lecture 2)	1 Hour
Module 2 (Linear Time Properties)		10 Hours
2.1	Linear-Time (LT) Properties - Deadlock	1 Hour
2.2	Linear-Time Behavior - Paths and State Graph, Path Fragment	1 Hour
2.3	Maximal and Initial Path Fragment, Path	
2.4	Traces - Trace and Trace Fragment	1 Hour
2.5	LT Property, Satisfaction Relation for LT Properties, Trace Equivalence and LT Properties	1 Hour
2.6	Invariants	1 Hour
2.7	Safety Properties, Trace Equivalence and Safety properties	1 Hour
2.8	Liveness Property, Safety vs. Liveness Properties	1 Hour
2.9	Fairness, Unconditional, Weak and Strong Fairness	1 Hour
2.10	Fairness Strategies, Fairness and Safety	1 Hour
Module 3 (Regular Properties)		8 Hours
3.1	Regular Properties - Model Checking Regular Safety properties - Regular Safety property	1 Hour
3.2	Verifying Regular Safety Properties	1 Hour

3.3	Automata on Infinite Words - ω -Regular Languages and Properties	1 Hour
3.4	Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA), Generalised Buchi Automata	1 Hour
3.5	Model Checking ω -Regular Properties - Persistence Properties and Product - Lecture 1	1 Hour
3.6	Persistence Properties and Product - Lecture 2	1 Hour
3.7	Nested Depth-First Search (Lecture 1)	1 Hour
3.8	Nested Depth-First Search (Lecture 2)	1 Hour
Module 4 (Linear Time Logic)		9 Hours
4.1	Linear Temporal Logic – Linear Temporal Logic (LTL) - Syntax	1 Hour
4.2	Semantics - Lecture 1	1 Hour
4.3	Semantics - Lecture 2	1 Hour
4.4	Equivalence of LTL Formulae, Weak Until	1 Hour
4.5	Release and Positive Normal Form	1 Hour
4.6	Fairness, Safety and Liveness in LTL	1 Hour
4.7	Automata Based LTL Model Checking - Lecture 1	1 Hour
4.8	Automata Based LTL Model Checking - Lecture 2	1 Hour
4.9	Automata Based LTL Model Checking - Lecture 3	1 Hour
Module 5 (Model Checking in SAL)		10 Hours
5.1	Introduction - Introduction to the tool Symbolic Analysis Laboratory (SAL).	1 Hour
5.2	The Language of SAL - The expression language, The transition Language	1 Hour
5.3	The module language, SAL Contexts.	1 Hour
5.4	SAL Examples - Mutual Exclusion	1 Hour
5.5	Peterson's Protocol	1 Hour
5.6	Synchronous Bus Arbiter	1 Hour
5.7	Bounded Bakery protocol,	1 Hour
5.8	Bakery Protocol	1 Hour

5.9	Simpson's Protocol	1 Hour
5.10	Stack	1 Hour



APJ ABDUL KALAM
TECHNOLOGICAL
UNIVERSITY

SEMESTER VI

KTU



CST 302	COMPILER DESIGN	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble:

The purpose of this course is to create awareness among students about the phases of a compiler and the techniques for designing a compiler. This course covers the fundamental concepts of different phases of compilation such as lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization and code generation. Students can apply this knowledge in design and development of compilers.

Prerequisite: Sound knowledge in Data Structures, Formal Languages & Automata Theory.

Course Outcomes: After the completion of the course the student will be able to

CO1	Explain the phases in compilation process (lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization and code generation) and model a lexical analyzer (Cognitive Knowledge Level: Apply)
CO2	Model language syntax using Context Free Grammar and develop parse tree representation using leftmost and rightmost derivations (Cognitive Knowledge Level: Apply)
CO3	Compare different types of parsers (Bottom-up and Top-down) and construct parser for a given grammar (Cognitive Knowledge Level: Apply)
CO4	Build Syntax Directed Translation for a context free grammar, compare various storage allocation strategies and classify intermediate representations (Cognitive Knowledge Level: Apply)
CO5	Illustrate code optimization and code generation techniques in compilation (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>
CO2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>
CO3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>
CO4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>
CO5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>

Abstract POs defined by National Board of Accreditation

PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Marks)	Test 2 (Marks)	
Remember	20	20	20
Understand	40	40	40
Apply	40	40	40
Analyze			

Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment - Test : **25 marks**

Continuous Assessment - Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction to compilers and lexical analysis)

Analysis of the source program - Analysis and synthesis phases, Phases of a compiler. Compiler writing tools. Bootstrapping. Lexical Analysis - Role of Lexical Analyser, Input Buffering, Specification of Tokens, Recognition of Tokens.

Module - 2 (Introduction to Syntax Analysis)

Role of the Syntax Analyser – Syntax error handling. Review of Context Free Grammars - Derivation and Parse Trees, Eliminating Ambiguity. Basic parsing approaches - Eliminating left recursion, left factoring. Top-Down Parsing - Recursive Descent parsing, Predictive Parsing, LL(1) Grammars.

Module - 3 (Bottom-Up Parsing)

Handle Pruning. Shift Reduce parsing. Operator precedence parsing (Concept only). LR parsing - Constructing SLR, LALR and canonical LR parsing tables.

Module - 4 (Syntax directed translation and Intermediate code generation)

Syntax directed translation - Syntax directed definitions, S-attributed definitions, L-attributed definitions, Bottom-up evaluation of S-attributed definitions. Run-Time Environments - Source Language issues, Storage organization, Storage-allocation strategies. Intermediate Code Generation - Intermediate languages, Graphical representations, Three-Address code, Quadruples, Triples.

Module 5 – (Code Optimization and Generation)

Code Optimization - Principal sources of optimization, Machine dependent and machine independent optimizations, Local and global optimizations. Code generation - Issues in the design of a code generator, Target Language, A simple code generator.

Text Books

1. Aho A.V., Ravi Sethi and D. Ullman. Compilers – Principles Techniques and Tools, Addison Wesley, 2006.

Reference Books

1. D.M.Dhamdhare, System Programming and Operating Systems, Tata McGraw Hill & Company, 1996.
2. Kenneth C. Loudon, Compiler Construction – Principles and Practice, Cengage Learning Indian Edition, 2006.

3. Tremblay and Sorenson, The Theory and Practice of Compiler Writing, Tata McGraw Hill & Company, 1984.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

- 1) Explain the phases of a compiler with a neat diagram.
- 2) Define a token. Identify the tokens in the expression $a := b + 10$.

Course Outcome 2 (CO2):

- 1) Illustrate the process of eliminating ambiguity, left recursion and left factoring the grammar.
- 2) Is the following grammar ambiguous? If so eliminate ambiguity.

$$E \rightarrow E + E \mid E * E \mid (E) \mid id$$

Course Outcome 3 (CO3):

1. What are the different parsing conflicts in the SLR parsing table?
2. Design a recursive descent parser for the grammar

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid id$$

3. Construct canonical LR(0) collection of items for the grammar below.

$$S \rightarrow L = R$$

$$S \rightarrow R$$

$$L \rightarrow * R$$

$$L \rightarrow id$$

$$R \rightarrow L$$

Also identify a shift reduce conflict in the LR(0) collection constructed above.

Course Outcome 4 (CO4):

1. Write the quadruple and triple representation of the following intermediate code

$$R1 = C * D$$

$$R2 = B + R1$$

$$A = R2$$

$$B[0] = A$$

2. Differentiate S-attributed Syntax Directed Translation(SDT) and L-attributed SDT. Write S - attributed SDT for a simple desktop calculator

Course Outcome 5 (CO5):

1. List out the examples of function preserving transformations.
2. What are the actions performed by a simple code generator for a typical three-address statement of the form $x: = y \text{ op } z$.

Model Question Paper

QP CODE:

Reg No: _____

Name: _____

PAGES : 4

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
SIXTH SEMESTER B.TECH DEGREE EXAMINATION , MONTH & YEAR**

Course Code: CST 302

Course Name: Compiler Design

**Max.Marks:100
Hours**

Duration: 3

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Specify the analysis and synthesis parts of compilation.
2. Define the terms token, lexemes and patterns with examples.
3. Is the grammar $S \rightarrow S \mid (S) S \mid \epsilon$ ambiguous? Justify your answer.
4. What is left recursive grammar? Give an example. What are the steps in removing left recursion?
5. Compare different bottom-up parsing techniques.
6. What are the possible actions of a shift reduce parser.

7. Differentiate synthesized and inherited attributes with examples.

8. Translate $a[i] = b * c - b * d$, to quadruple.

9. What is the role of peephole optimization in the compilation process

10. What are the issues in the design of a code generator

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the different phases of a compiler with a running example.

(9)

(b) List and explain any three compiler construction tools.

(5)

OR

12. (a) What is a regular definition? Give the regular definition of an unsigned integer

(7)

(b) Express the role of transition diagrams in recognition of tokens.

(7)

13. (a) What is Recursive Descent parsing? List the challenges in designing such a parser?

(4)

(b) Consider the following grammar

$E \rightarrow E \text{ or } T \mid T$

$T \rightarrow T \text{ and } F \mid F$

$F \rightarrow \text{not } F \mid (E) \mid \text{true} \mid \text{false}$

(10)

(i) Remove left recursion from the grammar.

(ii) Construct a predictive parsing table.

(iii) Justify the statement “The grammar is LL (1)”.

OR

14. (a) What is Recursive Descent parsing? List the problems in designing such a parser (4)

(b) Design a recursive descent parser for the grammar $S \rightarrow cAd, A \rightarrow ab/ b$ (5)

Find the FIRST and FOLLOW of the non-terminals S, A and B in the grammar (5)

$$S \rightarrow aABe$$

$$A \rightarrow Abc \mid b$$

$$B \rightarrow d$$

15. (a) Construct the LR(0) set of items and their GOTO function for the grammar $S \rightarrow S S + \mid S S * \mid a$ (10)

(b) Is the grammar SLR? Justify your answer (4)

OR

16. (a) Identify LR(1) items for the grammar (7)

$$S \rightarrow CC$$

$$C \rightarrow cC \mid d$$

(b) Construct LALR table for the above grammar (7)

17. (a) Design a Syntax Directed Translator(SDT) for the arithmetic expression $(4 * 7 + 19) * 2$ and draw an annotated parse tree for the same. (8)

(b) Consider the grammar with following translation rules and E as the start symbol (6)

$$E \rightarrow E1 \# T \{E.value = E1.value \times T.value ;\}$$

$$\mid T \{E.value = T.value ;\}$$

$$T \rightarrow T1 \& F \{ T.value = T1.value + F.value ;\}$$

$$\mid F \{ T.value = F.value ;\}$$

$$F \rightarrow num \{ F.value = num.lvalue ;\}$$

Compute E.value for the root of the parse tree for the expression

$2\#3 \& 5\# 6 \& 7$

OR

18. (a) Write Syntax Directed Translator (SDT) and parse tree for infix to postfix translation of an expression. (8)
- (b) Explain the storage allocation strategies. (6)
19. (a) Describe the principal sources of optimization (7)
- (b) Illustrate the optimization of basic blocks with examples. (7)

OR

20. (a) Write the Code Generation Algorithm and explain the *getreg* function (6)
- (b) Generate target code sequence for the following statement (8)
- $$d := (a-b) + (a-c) + (a-c).$$

Teaching Plan

No	Contents	No. of Lecture Hours
Module - 1(Introduction to Compilers and lexical analyzer) (8 hours)		
1.1	Introduction to compilers, Analysis of the source program	1 hour
1.2	Phases of the compiler – Analysis Phases	1 hour
1.3	Phases of the Compiler - Synthesis Phases	1 hour
1.4	Symbol Table Manager and Error Handler	1 hour
1.5	Compiler writing tools, bootstrapping	1 hour
1.6	The role of Lexical Analyzer , Input Buffering	1 hour
1.7	Specification of Tokens	1 hour
1.8	Recognition of Tokens	1 hour

Module – 2 (Introduction to Syntax Analysis) (10 hours)		
2.1	Role of the Syntax Analyser, Syntax error handling	1 hour
2.2	Review of Context Free Grammars	1 hour
2.3	Parse Trees and Derivations	1 hour
2.4	Grammar transformations, Eliminating ambiguity	1 hour
2.5	Eliminating left recursion	1 hour
2.6	Left factoring the grammar	1 hour
2.7	Recursive Descent parsing	1 hour
2.8	First and Follow	1 hour
2.9	Predictive Parsing table constructor	1 hour
2.10	LL(1) Grammars	1 hour
Module - 3 (Bottom up parsing) (9 hours)		
3.1	Bottom-up parsing - Handle Pruning	1 hour
3.2	Shift Reduce parsing	1 hour
3.3	Operator precedence parsing (Concept only)	1 hour
3.4	LR parsing , SLR Grammar, items	1 hour
3.5	Augmented Grammar, Canonical collection of LR(0) items	1 hour
3.6	SLR Parser Table Construction	1 hour
3.7	Constructing Canonical LR Parsing Tables	1 hour
3.8	Constructing LALR Parsing Tables	1 hour
3.9	LALR parser	1 hour
Module - 4 (Syntax Directed Translation and Intermediate code Generation) (9 hours)		
4.1	Syntax directed definitions	1 hour
4.2	S- attributed definitions, L- attributed definitions	1 hour
4.3	Bottom- up evaluation of S- attributed definitions.	1 hour
4.4	Source Language issues	1 hour
4.5	Storage organization	1 hour

4.6	Storage- allocation strategies	1 hour
4.7	Intermediate languages , Graphical representations	1 hour
4.8	Three-Address code	1 hour
4.9	Quadruples, Triples	1 hour
Module - 5 (Code Optimization and Generation) (9 hours)		
5.1	Principal sources of optimization	1 hour
5.2	Machine dependent optimizations	1 hour
5.3	Machine independent optimizations	1 hour
5.4	Local optimizations	1 hour
5.5	Global optimizations	1 hour
5.6	Issues in the design of a code generator – Lecture 1	1 hour
5.7	Issues in the design of a code generator – Lecture 2	1 hour
5.8	Target Language	1 hour
5.9	Design of a simple code generator.	1 hour

CST 304	COMPUTER GRAPHICS AND IMAGE PROCESSING	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble:

The purpose of this course is to make awareness about strong theoretical relationships between computer graphics and image processing. This course helps the learner to understand three-dimensional environment representation in a computer, transformation of 2D/3D objects, basic mathematical techniques and algorithms used to build useful applications, imaging, and image processing techniques. The study of computer graphics and image processing develops the ability to create image processing frameworks for different domains and develops algorithms for emerging display technologies.

Prerequisite: A sound knowledge of Mathematics and a programming language.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Describe the working principles of graphics devices(Cognitive Knowledge level: Understand)
CO2	Illustrate line drawing, circle drawing and polygon filling algorithms(Cognitive Knowledge level: Apply)
CO3	Demonstrate geometric representations, transformations on 2D & 3D objects, clipping algorithms and projection algorithms(Cognitive Knowledge level: Apply)
CO4	Summarize visible surface detection methods(Cognitive Knowledge level: Understand)
CO5	Summarize the concepts of digital image representation, processing and demonstrate pixel relationships(Cognitive Knowledge level: Apply)
CO6	Solve image enhancement and segmentation problems using spatial domain techniques(Cognitive Knowledge level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	<input checked="" type="checkbox"/>											<input checked="" type="checkbox"/>
CO2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>
CO3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>
CO4	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>									<input checked="" type="checkbox"/>
CO5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								
CO6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30

Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	10 marks
Continuous Assessment Tests (Average of Series Tests 1 & 2)	25 marks
Continuous Assessment Assignment	15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one full question. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1(Basics of Computer graphics and Algorithms)

Basics of Computer Graphics and its applications. Video Display devices- Refresh Cathode Ray Tubes, Random Scan Displays and systems, Raster scan displays and systems. Line drawing algorithms- DDA, Bresenham's algorithm. Circle drawing algorithms- Midpoint Circle generation algorithm, Bresenham's algorithm.

Module - 2(Filled Area Primitives and transformations)

Filled Area Primitives- Scan line polygon filling, Boundary filling and flood filling. Two dimensional transformations-Translation, Rotation, Scaling, Reflection and Shearing, Composite transformations, Matrix representations and homogeneous coordinates. Basic 3D transformations.

Module - 3 (Clipping and Projections)

Window to viewport transformation. Cohen Sutherland Line clipping algorithm. Sutherland Hodgeman Polygon clipping algorithm. Three dimensional viewing pipeline. Projections- Parallel and Perspective projections. Visible surface detection algorithms- Depth buffer algorithm, Scan line algorithm.

Module - 4 (Fundamentals of Digital Image Processing)

Introduction to Image processing and applications. Image as 2D data. Image representation in Gray scale, Binary and Colour images. Fundamental steps in image processing. Components of image processing system. Coordinate conventions. Sampling and quantization. Spatial and Gray Level Resolution. Basic relationship between pixels- neighbourhood, adjacency, connectivity. Fundamentals of spatial domain-convolution operation.

Module - 5 (Image Enhancement in Spatial Domain and Image Segmentation)

Basic gray level transformation functions - Log transformations, Power-Law transformations, Contrast stretching. Histogram equalization. Basics of spatial filtering - Smoothing spatial filter- Linear and nonlinear filters, and Sharpening spatial filters-Gradient and Laplacian.

Fundamentals of Image Segmentation. Thresholding - Basics of Intensity thresholding and Global Thresholding. Region based Approach - Region Growing, Region Splitting and Merging. Edge Detection - Edge Operators- Sobel and Prewitt.

Text Book

1. Donald Hearn and M. Pauline Baker, Computer Graphics, PHI, 2e, 1996
2. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing. Pearson, 4e, 2017

References

- 1) William M. Newman and Robert F. Sproull, Principles of Interactive Computer Graphics. McGraw Hill, 2001

- 2) Zhigang Xiang and Roy Plastock, Computer Graphics (Schaum's outline Series), McGraw Hill, 2019.
- 3) David F. Rogers , Procedural Elements for Computer Graphics, Tata McGraw Hill,2001.
- 4) M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, Thomson India Edition, 4e, 2017.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare the working principle of raster scan systems and random scan systems.
2. How much time is spent scanning across each row of pixels during screen refresh on a raster system with resolution of 1280*1024 and a refresh rate of 60 frames per second?

Course Outcome 2 (CO2):

1. Rasterize the line with end points(2,3) and (5,8) using Bresenham's line drawing algorithm.
2. Explain how the 4-connected area filling approach differs from 8- connected area filling in boundary filling algorithm

Course Outcome 3 (CO3):

1. Rotate a triangle ABC 45 degree counter clockwise about the pivot point (10,3), where the position vector of the coordinate ABC is given as A(4,1), B(5,2) and C(4,3).
2. Given a clipping window A(20,20), B(60,20), C(60,40) and D(20,40). Using Cohen Sutherland algorithm, find the visible portion of the line segment joining the points P(40,80) and Q(120,30)

Course Outcome 4 (CO4):

1. Explain scan line algorithm for detecting visible surfaces in an object.

Course Outcome 5 (CO5):

1. Give an image representation model and describe how the representation changes in grayscale, binary and colour images.
2. Consider an image segment shown below.

3 1 2 1 (q)

2 2 0 2

1 2 1 1

(p) 1 0 1 2

- (a) Let $V=\{0,1\}$ and compute the length of the shortest 4-,8- and m- path between p and q. If a particular path does not exist between these two points , explain why?
- (b) Repeat for $V=\{1,2\}$.

3. The spatial resolution of an image is given by 128 X 128. What are its storage requirements if it is represented by 64 gray levels?

Course Outcome 6 (CO6):

1. A skilled medical technician is charged with the job of inspecting a certain class of monochrome images generated by electronic microscope. To facilitate the inspection, the technician uses image processing aids. However when he examines the images he finds the following problems.
- Presence of bright isolated dots that are not of interest.
 - Lack of sharpness
 - Poor contrast

Identify the sequence of preprocessing steps that the technician may use to overcome the above mentioned problems and explain it.

2. A 4x4, 4 bits/pixel original image is given by

{	10	12	8	9
	10	12	12	14
	12	13	10	9
	14	12	10	12

- Apply histogram equalisation to the image by rounding the resulting image pixels to integers
 - Sketch the histogram of the original image and the histogram-equalised image.
3. You have Sobel operator and Laplacian operator for edge detection. Which operator will you select for edge detection in the case of noisy image? Explain. **(Assignment)**

Model Question Paper

QP CODE:

Reg No: _____

Name: _____

PAGES : 4

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 304

Course Name: Computer Graphics and Image Processing

Max. Marks : 100

Duration: 3 Hours

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Justify the approach of using integer arithmetic in Bresenham's line drawing algorithm.
2. Consider a raster system with a resolution of 1024×1024 . What is the size of the raster needed to store 4 bits per pixel? How much storage is needed if 8 bits per pixel are to be stored?
3. Show that two successive reflections about either of the coordinate axes is equivalent to a single rotation about the coordinate origin.
4. Determine a sequence of basic transformations that are equivalent to the x-direction shearing matrix.
5. Find the window to viewport normalization transformation with window lower left corner at (1,1) and upper right corner at (2,6).
6. Find the orthographic projection of a unit cube onto the $x=0$, $y=0$ and $z=0$ plane.
7. Define Sampling and Quantization of an image.

8. Give any three applications of digital image processing.
9. A captured image appears very dark because of wrong lens aperture setting. Describe an enhancement technique which is appropriate to enhance such an image.
10. Suggest an approach of thresholding that should be used in case of uniform illumination. (10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Write Midpoint circle drawing algorithm and use it to plot a circle with radius=20 and center is (50,30). (10)
- (b) Draw the architecture of raster scan display systems and explain its working principle. (4)
- OR**
12. (a) Derive the initial decision parameter of Bresenham's line drawing algorithm and use the algorithm to rasterize a line with endpoints (2,2) and (10,10). (10)
- (b) Explain the working principle of color CRT monitors with suitable illustrations. (4)
13. (a) Compare boundary fill algorithm and flood fill algorithm. (5)
- (b) Reflect a triangle ABC about the line $3x-4y+8=0$. The position vector of the coordinate ABC is given as A(4,1), B(5,2) and C(4,3). (9)
- OR**
14. (a) Explain the need of using vanishing points in projections. (4)
- (b) Explain Cohen-Sutherland line clipping algorithm. Use the algorithm to clip line P1(70, 20) and P2(100,10) against a window lower left hand corner (50,10) and upper right hand corner (80,40). (10)
15. (a) Describe Sutherland Hodegman polygon clipping algorithm and what are its (7)

limitations.

- (b) Explain how visible surfaces can be detected using depth buffer algorithm. (7)

OR

16. (a) Describe Sutherland Hodegman polygon clipping algorithm and what are its limitations. (7)

- (b) Explain how visible surfaces can be detected using depth buffer algorithm. (7)

17. (a) Explain the components of an image processing system with suitable diagram (9)

- (b) Define Resolution of an image. Explain the spatial and gray level resolution of an image with an example. (5)

OR

18. (a) Define 4-adjacency, 8 adjacency and m-adjacency. Consider the image segment shown. (7)

4 2 3 2 (q)
3 3 1 3
2 3 2 2

(p) 2 1 2 3

Let $V=\{1,2\}$ and compute the length of the shortest 4-,8- and m- path between p and q. If a particular path does not exist between these two points, explain why?

- (b) Using any one application, explain the steps involved in image processing. (7)

19. (a) A 5x5 image patch is shown below. Compute the value of the marked pixel if it is smoothened by a 3x3 average filter and median filter. (4)

$$f(m,n) = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 \\ 5 & 6 & 7 & 8 & 4 \\ 4 & 3 & \textcircled{2} & 1 & 2 \\ 8 & 7 & 6 & 5 & 3 \\ 1 & 5 & 3 & 7 & 6 \end{pmatrix}$$

- (b) Define Image segmentation and describe in detail method of edge and region based segmentation technique. (10)

OR

20. (a) Distinguish between smoothing and sharpening filters in terms of (10)
- (i) Functionality
 - (ii) Types
 - (iii) Applications
 - (iv) Mask Coefficients
- (b) Describe how an image is segmented using split and merge technique in association with the region adjacency graph. (8)

Teaching Plan

No	Contents	No of Lecture Hrs (45 hrs)
Module – 1 (Basics of Computer Graphics and Algorithms) (9 hrs)		
1.1	Basics of Computer Graphics and applications	1 hour
1.2	Refresh Cathode Ray Tubes	1 hour
1.3	Random Scan Displays and systems	1 hour
1.4	Raster scan displays and systems	1 hour
1.5	DDA Line drawing Algorithm	1 hour
1.6	Bresenham's line drawing algorithm	1 hour
1.7	Midpoint Circle generation algorithm	1 hour
1.8	Bresenham's Circle generation algorithm	1 hour
1.9	Illustration of line drawing and circle drawing algorithms	1 hour
Module - 2 (Filled Area Primitives and transformations) (9 hrs)		
2.1	Scan line polygon filling	1 hour
2.2	Boundary filling and flood filling	1 hour
2.3	Basic 2D transformations-Translation	1 hour

2.4	Basic 2D transformations- Rotation and Scaling	1 hour
2.5	Reflection and Shearing	1 hour
2.6	Composite transformations	1 hour
2.7	Matrix representations and homogeneous coordinates	1 hour
2.8	Basic 3D transformation-Translation and scaling	1 hour
2.9	Basic 3D transformation-Rotation	1 hour
Module - 3 (Clipping and Projections) (8 hrs)		
3.1	Window to viewport transformation	1 hour
3.2	Cohen Sutherland Line clipping algorithm	1 hour
3.3	Sutherland Hodgeman Polygon clipping algorithm	1 hour
3.4	Practice problems on Clipping algorithms	1 hour
3.5	Three dimensional viewing pipeline, Projections-Parallel projections	1 hour
3.6	Projections- Perspective projections	1 hour
3.7	Visible surface detection algorithms- Depth buffer algorithm	1 hour
3.8	Scan line visible surface detection algorithm	1 hour
Module - 4 (Fundamentals of Digital Image Processing) (8 hrs)		
4.1	Introduction to Image processing-Image as a 2D data, Image representation-Gray scale, Binary and Colour images.	1 hour
4.2	Fundamental steps in image processing and applications	1 hour
4.3	Components of image processing system	1 hour
4.4	Coordinate conventions, Sampling and quantization, Spatial and Gray Level Resolution	1 hour
4.5	Basic relationship between pixels – neighbourhood, adjacency, connectivity	1 hour
4.6	Illustration of basic relationship between pixels– neighbourhood,	1 hour

	adjacency, connectivity	
4.7	Fundamentals of spatial domain - Convolution operation	1 hour
4.8	Illustration of Convolution operation	1 hour
Module - 5 (Image Enhancement in spatial domain and Image Segmentation) (11 hrs)		
5.1	Basic gray level transformation functions- Log transformations.	1 hour
5.2	Power-Law transformations, Contrast stretching	1 hour
5.3	Histogram equalization	1 hour
5.4	Illustration of Histogram equalization	1 hour
5.5	Basics of spatial filtering, Smoothing spatial filter- Linear and nonlinear filters	1 hour
5.6	Sharpening spatial filtering-Gradient filter mask	1 hour
5.7	Sharpening spatial filtering-Laplacian filter mask	1 hour
5.8	Fundamentals of Image Segmentation, Basics of Intensity thresholding, Basic Global Thresholding	1 hour
5.9	Region Based Approach- Region Growing, Region Splitting and Merging	1 hour1
5.10	Basics of Edge Detection	1 hour
5.11	Sobel and Prewitt edge detection masks	1 hour

CST 306	ALGORITHM ANALYSIS AND DESIGN	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble:

The course introduces students to the design of computer algorithms, as well as analysis of algorithms. Algorithm design and analysis provide the theoretical backbone of computer science and are a must in the daily work of the successful programmer. The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. At the end of the course students will be able to develop their own versions for a given computational task and to compare and contrast their performance.

Prerequisite:

Strong Foundation in Mathematics, Programming in C, Data Structures and Graph Theory.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Analyze any given algorithm and express its time and space complexities in asymptotic notations. (Cognitive Level: Apply)
CO2	Derive recurrence equations and solve it using Iteration, Recurrence Tree, Substitution and Master's Method to compute time complexity of algorithms. (Cognitive Level: Apply)
CO3	Illustrate Graph traversal algorithms & applications and Advanced Data structures like AVL trees and Disjoint set operations. (Cognitive Level: Apply)
CO4	Demonstrate Divide-and-conquer, Greedy Strategy, Dynamic programming, Branch-and Bound and Backtracking algorithm design techniques (Cognitive Level: Apply)
CO5	Classify a problem as computationally tractable or intractable, and discuss strategies to address intractability (Cognitive Level: Understand)
CO6	Identify the suitable design strategy to solve a given problem. (Cognitive Level: Analyze)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	☑	☑	☑	☑								☑
CO2	☑	☑	☑	☑								☑
CO3	☑	☑	☑	☑								☑
CO4	☑	☑	☑	☑								☑
CO5	☑	☑										√
CO6	☑	☑	☑	☑								☑

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40

Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	10 marks
Continuous Assessment Tests (Average of Series Tests 1 & 2)	25 marks
Continuous Assessment Assignment	15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Introduction to Algorithm Analysis)

Characteristics of Algorithms, Criteria for Analysing Algorithms, Time and Space Complexity - Best, Worst and Average Case Complexities, Asymptotic Notations - Big-Oh (O), Big- Omega (Ω), Big-Theta (Θ), Little-oh (o) and Little- Omega (ω) and their properties. Classifying functions by their asymptotic growth rate, Time and Space Complexity Calculation of simple algorithms.

Analysis of Recursive Algorithms: Recurrence Equations, Solving Recurrence Equations – Iteration Method, Recursion Tree Method, Substitution method and Master’s Theorem (Proof not required).

Module–2 (Advanced Data Structures and Graph Algorithms)

Self Balancing Tree - AVL Trees (Insertion and deletion operations with all rotations in detail, algorithms not expected); Disjoint Sets- Disjoint set operations, Union and find algorithms.

DFS and BFS traversals - Analysis, Strongly Connected Components of a Directed graph, Topological Sorting.

Module–3 (Divide & Conquer and Greedy Strategy)

The Control Abstraction of Divide and Conquer- 2-way Merge sort, Strassen’s Algorithm for Matrix Multiplication-Analysis. The Control Abstraction of Greedy Strategy- Fractional Knapsack Problem, Minimum Cost Spanning Tree Computation- Kruskal’s Algorithms - Analysis, Single Source Shortest Path Algorithm - Dijkstra’s Algorithm-Analysis.

Module-4 (Dynamic Programming, Back Tracking and Branch & Bound))

The Control Abstraction- The Optimality Principle- Matrix Chain Multiplication-Analysis, All Pairs Shortest Path Algorithm - Floyd-Warshall Algorithm-Analysis. The Control Abstraction of Back Tracking – The N Queen’s Problem. Branch and Bound Algorithm for Travelling Salesman Problem.

Module-5 (Introduction to Complexity Theory)

Tractable and Intractable Problems, Complexity Classes – P, NP, NP- Hard and NP-Complete Classes- NP Completeness proof of Clique Problem and Vertex Cover Problem- Approximation algorithms- Bin Packing, Graph Coloring. Randomized Algorithms (Definitions of Monte Carlo and Las Vegas algorithms), Randomized version of Quick Sort algorithm with analysis.

Text Books

1. T.H.Cormen, C.E.Leiserson, R.L.Rivest, C. Stein, Introduction to Algorithms, 2nd Edition, Prentice-Hall India (2001)
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, 2nd Edition, Orient Longman Universities Press (2008)

3. Sara Baase and Allen Van Gelder —Computer Algorithms, Introduction to Design and Analysis, 3rd Edition, Pearson Education (2009)

Reference Books

1. Jon Kleinberg, Eva Tardos, “Algorithm Design”, First Edition, Pearson (2005)
2. Robert Sedgewick, Kevin Wayne, “Algorithms”, 4th Edition Pearson (2011)
3. Gilles Brassard, Paul Bratley, “Fundamentals of Algorithmics”, Pearson (1996)
4. Steven S. Skiena, “The Algorithm Design Manual”, 2nd Edition, Springer(2008)

Course Level Assessment Questions

Course Outcome 1 (CO1):

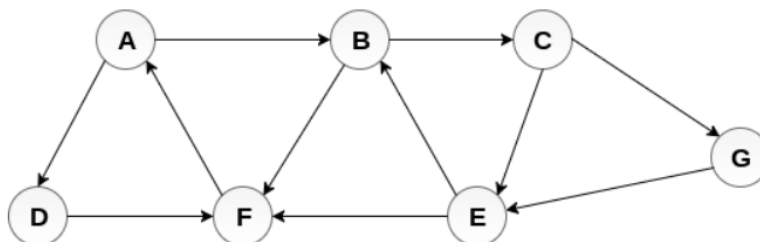
1. Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$? Justify your answer.
2. What is the need of asymptotic analysis in calculating time complexity? What are the notations used for asymptotic analysis?
3. Calculate the time complexity for addition of two matrices.
4. Define time complexity and space complexity. Write an algorithm for adding n natural numbers and analyse the time and space requirements of the algorithm.

Course Outcome 2 (CO2):

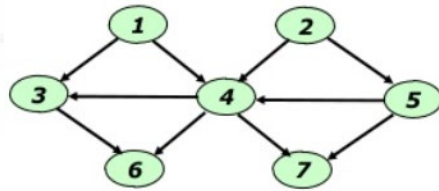
1. State Master’s theorem for solving recurrences.
2. Solve the recurrence $T(n) = 3T(n-2)$, using iteration method
3. State the conditions in recurrences where Master Theorem is not applicable.
4. Solve the following recurrence equations using Master’s theorem.
 - a) $T(n) = 8T(n/2) + 100n^2$
 - b) $T(n) = 2T(n/2) + 10n$
5. Using Recursion Tree method, Solve $T(n) = 2T(n/10) + T(9n/10) + n$. Assume constant time for small values of n.

Course Outcome 3 (CO3):

1. Explain the rotations performed for insertion in AVL tree with example.
2. Write down BFS algorithm and analyse the time complexity. Perform BFS traversal on the given graph starting from node A. If multiple node choices are available for next travel, choose the next node in alphabetical order.

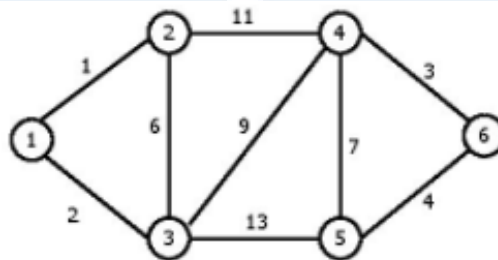


- Find the minimum and maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0. (3)
- Find any three topological orderings of the given graph.

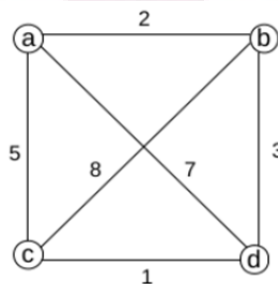


Course Outcome 4 (CO4):

- Give the control abstraction for Divide and Conquer method.
- Construct the minimum spanning tree for the given graph using Kruskal’s algorithm. Analyse the complexity of the algorithm.



- Compare Divide and Conquer and Dynamic programming methodologies
- What is Principle of Optimality?
- Define Travelling Salesman Problem (TSP). Apply branch and bound algorithm to solve TSP for the following graph, assuming the start city as ‘a’. Draw the state space tree.



Course Outcome 5 (CO5):

- Compare Tractable and Intractable Problems
- With the help of suitable code sequence convince Vertex Cover Problem is an example of NP-Complete Problem

3. Explain Vertex Cover problem using an example. Suggest an algorithm for finding Vertex Cover of a graph.
4. Write short notes on approximation algorithms.
5. Compare Conventional quick sort algorithm and Randomized quicksort with the help of a suitable example?

Course Outcome 6 (CO6): (CO attainment through assignment only, not meant for examinations)

Choosing the best algorithm design strategy for a given problem after applying applicable design strategies – Sample Problems Given.

1. Finding the Smallest and Largest elements in an array of 'n' numbers
2. Fibonacci Sequence Generation.
3. Merge Sort
4. Travelling Sales Man Problem
5. 0/1 Knapsack Problem

Model Question Paper

QP CODE:

Reg No: _____

Name: _____

PAGES : 4

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 306

Course Name: Algorithm Analysis and Design

Max. Marks : 100

Duration: 3 Hours

PART A

Answer All Questions. Each Question Carries 3 Marks

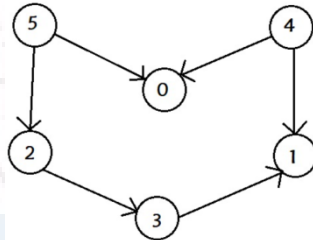
1. Define asymptotic notation? Arrange the following functions in increasing order of asymptotic growth rate.
 $n^3, 2^n, \log n^3, 2^{100}, n^2 \log n, n^n, \log n, n^{0.3}, 2^{\log n}$

2. State Master's Theorem. Find the solution to the following recurrence equations using Master's theorem.

a) $T(n) = 8T(n/2) + 100n^2$

b) $T(n) = 2T(n/2) + 10n$

3. Find any two topological ordering of the DAG given below.



4. Show the UNION operation using linked list representation of disjoint sets.
5. Write the control abstraction of greedy strategy to solve a problem.
6. Write an algorithm based on divide-and-conquer strategy to search an element in a given list. Assume that the elements of list are in sorted order.
7. List the sequence of steps to be followed in Dynamic Programming approach.
8. Illustrate how optimal substructure property could be maintained in Floyd-Warshall algorithm.
9. Differentiate between P and NP problems.
10. Specify the relevance of approximation algorithms.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Define Big O, Big Ω and Big Θ Notation and illustrate them graphically. (7)
- (b) Solve the following recurrence equation using recursion tree method (7)
- $T(n) = T(n/3) + T(2n/3) + n$, where $n > 1$
- $T(n) = 1$, Otherwise

OR

12. (a) Explain the iteration method for solving recurrences and solve the following recurrence equation using iteration method. (7)

$$T(n) = 3T(n/3) + n; T(1) = 1$$

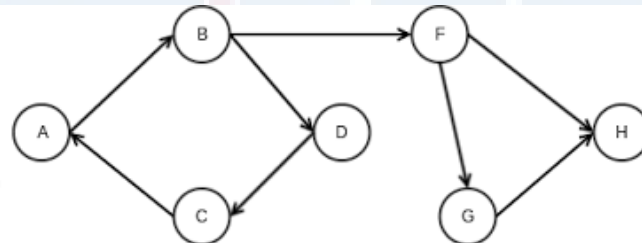
- (b) Determine the time complexities of the following two functions fun1() and fun2(). (7)

```
i) int fun1(int n)
{
    if (n <= 1) return n;
    return 2*fun1(n-1);
}
```

```
ii) int fun2 (int n)
{
    if (n <= 1) return n;
    return fun2 (n-1) + fun2 (n-1)
}
```

13. (a) Write DFS algorithm and analyse its time complexity. Illustrate the classification of edges in DFS traversal. (7)

- (b) Find the strongly connected components of the digraph given below: (7)



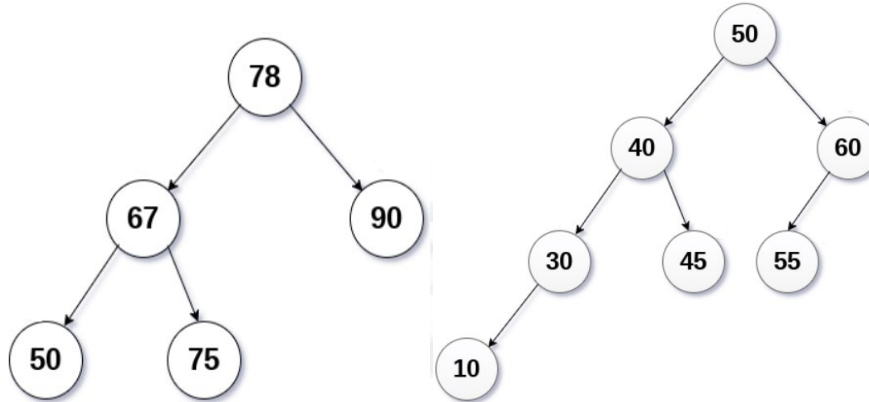
OR

14. (a) Illustrate the advantage of height balanced binary search trees over binary search trees? Explain various rotations in AVL trees with example. (7)

- (b) Perform the following operations in the given AVL trees. (7)

i) Insert 70

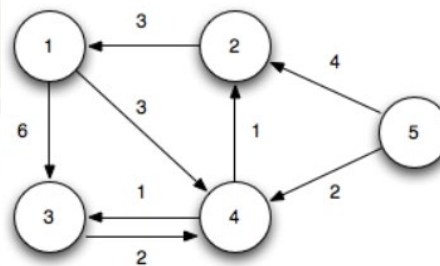
ii) Delete 55



15. (a) State Fractional Knapsack Problem and write Greedy Algorithm for Fractional Knapsack Problem. (7)
- (b) Find the optimal solution for the following Fractional Knapsack problem. (7)
 Given the number of items(n) = 7, capacity of sack(m) = 15,
 $W = \{2, 3, 5, 7, 1, 4, 1\}$ and $P = \{10, 5, 15, 7, 6, 18, 3\}$

OR

16. (a) Write and explain merge sort algorithm using divide and conquer strategy using the data $\{30, 19, 35, 3, 9, 46, 10\}$. Also analyse the time complexity. (7)
- (b) Write the pseudo code for Dijkstra's algorithm. Compute the shortest distance from vertex 1 to all other vertices using Dijkstra's algorithm. (7)

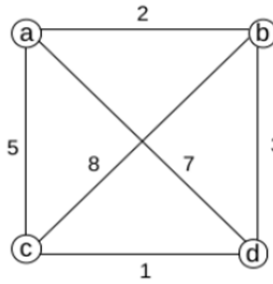


17. (a) Write Floyd-Warshall algorithm and analyse its complexity. (5)
- (b) Write and explain the algorithm to find the optimal parenthesization of matrix chain product whose sequence of dimension is $4 \times 10, 10 \times 3, 3 \times 12, 12 \times 20$. (9)

OR

18. (a) Explain the concept of Backtracking method using 4 Queens problem. (7)

- (b) Define Travelling Salesman Problem (TSP). Apply branch and bound algorithm to solve TSP for the following graph, assuming the start city as 'a'. Draw the state space tree. (7)



19. (a) State bin packing problem? Explain the first fit decreasing strategy (7)

- (b) Prove that the Clique problem is NP-Complete. (7)

OR

20. (a) Explain the need for randomized algorithms. Differentiate Las Vegas and Monte Carlo algorithms. (6)

- (b) Explain randomized quicksort and analyse the expected running time of randomized quicksort with the help of a suitable example? (9)

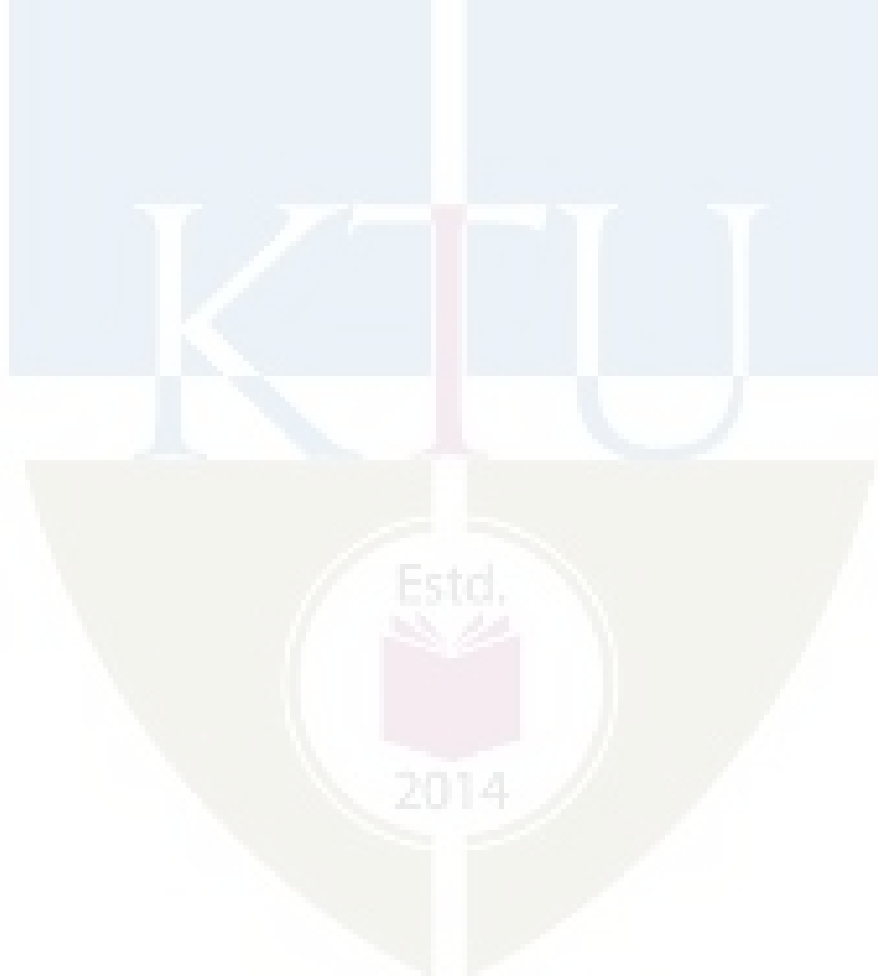
Teaching Plan

No	Topic	No. of Hours (45 hrs)
Module -1 (Introduction to Algorithm Analysis) 9 hrs.		
1.1	Introduction to Algorithm Analysis: Characteristics of Algorithms.	1 hour
1.2	Criteria for Analysing Algorithms, Time and Space Complexity - Best, Worst and Average Case Complexities.	1 hour
1.3	Asymptotic Notations - Properties of Big-Oh (O), Big- Omega (Ω), Big-Theta (Θ), Little-Oh (o) and Little- Omega (ω).	1 hour
1.4	Illustration of Asymptotic Notations	1 hour

1.5	Classifying functions by their asymptotic growth rate	1 hour
1.6	Time and Space Complexity Calculation of algorithms/code segments.	1 hour
1.7	Analysis of Recursive Algorithms: Recurrence Equations, Solving Recurrence Equations – Iteration Method.	1 hour
1.8	Recursion Tree Method	1 hour
1.9	Substitution method and Master’s Theorem and its Illustration.	1 hour
Module-2 (Advanced Data Structures and Graph Algorithms) 10 Hrs.		
2.1	Self Balancing Trees - Properties of AVL Trees, Rotations of AVL Trees	1 hour
2.2	AVL Trees Insertion and Illustration	1 hour
2.3	AVL Trees Deletion and Illustration	1 hour
2.4	Disjoint set operations.	1 hour
2.5	Union and find algorithms.	1 hour
2.6	Illustration of Union and find algorithms	1 hour
2.7	Graph Algorithms: BFS traversal, Analysis.	1 hour
2.8	DFS traversal, Analysis.	1 hour
2.9	Strongly connected components of a Directed graph.	1 hour
2.10	Topological Sorting.	1 hour
Module-3 (Divide & Conquer and Greedy Method) 8 Hrs		
3.1	Divide and Conquer: The Control Abstraction.	1 hour
3.2	2-way Merge Sort, Analysis.	1 hour
3.3	Strassen’s Algorithm for Matrix Multiplication, Analysis	1 hour

3.4	Greedy Strategy: The Control Abstraction.	1 hour
3.5	Fractional Knapsack Problem.	1 hour
3.6	Minimum Cost Spanning Tree Computation- Kruskal's Algorithm, Analysis.	1 hour
3.7	Single Source Shortest Path Algorithm - Dijkstra's Algorithm	1 hour
3.8	Illustration of Dijkstra's Algorithm-Analysis.	1 hour
Module-4 (Dynamic Programming, Back Tracking and Branch and Bound) 8 Hrs.		
4.1	Dynamic Programming: The Control Abstraction, The Optimality Principle.	1 hour
4.2	Matrix Chain Multiplication-Analysis.	1 hour
4.3	Illustration of Matrix Chain Multiplication-Analysis.	1 hour
4.4	All Pairs Shortest Path Algorithm- Analysis and Illustration of Floyd-Warshall Algorithm.	1 hour
4.5	Back Tracking: The Control Abstraction .	1 hour
4.6	Back Tracking: The Control Abstraction – The N Queen's Problem.	1 hour
4.7	Branch and Bound:- Travelling salesman problem.	1 hour
4.8	Branch and Bound:- Travelling salesman problem.	1 hour
Module-5 (Introduction to Complexity Theory) 10 Hrs		
5.1	Introduction to Complexity Theory: Tractable and Intractable Problems.	1 hour
5.2	Complexity Classes – P, NP.	1 hour
5.3	NP- Hard and NP-Complete Problems.	1 hour
5.4	NP Completeness Proof of Clique Problem.	1 hour

5.5	NP Completeness Proof of Vertex Cover Problem.	1 hour
5.6	Approximation algorithms- Bin Packing Algorithm and Illustration.	1 hour
5.7	Graph Colouring Algorithm and Illustration.	1 hour
5.8	Randomized Algorithms (definitions of Monte Carlo and Las Vegas algorithms).	1 hour
5.9	Randomized Version of Quick Sort Algorithm with Analysis.	1 hour
5.10	Illustration of Randomized Version of Quick Sort Algorithm with Analysis.	1 hour



CST 308	COMPREHENSIVE COURSE WORK	Category	L	T	P	Credit	Year of Introduction
		PCC	1	0	0	1	2019

Preamble:

The objective of this Course work is to ensure the comprehensive knowledge of each student in the most fundamental core courses in the curriculum. Six core courses credited from Semesters 3, 4 and 5 are chosen for the detailed study in this course work. This course helps the learner to become competent in cracking GATE, placement tests and other competitive examinations

Prerequisite:

1. Discrete Mathematical Structures
2. Data Structures
3. Operating Systems
4. Computer Organization And Architecture
5. Database Management Systems
6. Formal Languages And Automata Theory

Course Outcomes: After the completion of the course the student will be able to

CO1	Comprehend the concepts of discrete mathematical structures (Cognitive Knowledge Level: Understand)
CO2 :	Comprehend the concepts and applications of data structures (Cognitive Knowledge Level: Understand)
CO3 :	Comprehend the concepts, functions and algorithms in Operating System (Cognitive Knowledge Level: Understand)
CO4 :	Comprehend the organization and architecture of computer systems (Cognitive Knowledge Level: Understand)
CO5 :	Comprehend the fundamental principles of database design and manipulation (Cognitive Knowledge Level: Understand)
CO6 :	Comprehend the concepts in formal languages and automata theory Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>										<input checked="" type="checkbox"/>
CO2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>										<input checked="" type="checkbox"/>
CO3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>										<input checked="" type="checkbox"/>
CO4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>										<input checked="" type="checkbox"/>
CO5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>										<input checked="" type="checkbox"/>

Assessment Pattern

Bloom's Category	End Semester Examination
Remember	10
Understand	20
Apply	20
Analyse	
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
50	0	50	1 hour

End Semester Examination Pattern: Objective Questions with multiple choice (Four). Question paper include fifty questions of one mark each covering the five identified courses.

Syllabus

Full Syllabus of all six selected Courses.

1. Discrete Mathematical Structures
2. Data Structures
3. Operating Systems
4. Computer Organization And Architecture
5. Database Management Systems
6. Formal Languages And Automata Theory

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	DISCRETE MATHEMATICAL STRUCTURES (14 hours)	
1.1	Mock Test on Module 1 and Module 2	1 hour
1.2	Mock Test on Module 3, Module 4 and Module 5	1 hour
2	DATA STRUCTURES	
2.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
2.2	Mock Test on Module 4 and Module 5	1 hour
3	OPERATING SYSTEMS	
3.1	Mock Test on Module 1 and Module 2	1 hour
3.2	Mock Test on Module 3, Module 4 and Module 5	1 hour
3.3	Feedback and Remedial	1 hour
4	COMPUTER ORGANIZATION AND ARCHITECTURE	
4.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
4.2	Mock Test on Module 4 and Module 5	1 hour
5	DATABASE MANAGEMENT SYSTEMS	

5.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
5.2	Mock Test on Module 4 and Module 5	1 hour
6	FORMAL LANGUAGES AND AUTOMATA THEORY	
6.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
6.2	Mock Test on Module 4 and Module 5	1 hour
6.3	Feedback and Remedial	1 hour

Model Question Paper

QP CODE:

Reg No: _____

Name: _____

PAGES : 10

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR**

Course Code: CST 308

Course Name: Comprehensive Course Work

Max. Marks: 50

Duration: 1 Hour

**Objective type questions with multiple choices. Mark one correct answer for each question.
Each Question Carries 1 Mark**

- What is the maximum possible number of relations from a set with 5 elements to another set with 4 elements?
(A) 2^{10} (B) 2^{16} (C) 2^{20} (D) 2^{25}
- The set $\{1,2,4,7,8,11,13,14\}$ is a group under multiplication modulo 15. Find the inverse of element 13
(A) 7 (B) 13 (C) 1 (D) 8
- Consider the recurrence relation $a_1 = 2, a_n = 3n + a_{n-1}$ Then a_{72} is

- (A) 7882 (B) 7883 (C) 7884 (D) 7885
4. Which among the following is a contradiction?
 (A) $(p \wedge q) \vee \neg(p \vee q)$ (B) $(p \vee q) \wedge \neg(p \wedge q)$
 (C) $(p \wedge q) \wedge \neg(p \vee q)$ (D) $(p \wedge q) \vee (p \wedge \neg q)$
5. The number of non-negative solutions to $x + y + z = 18$, with conditions $x \geq 3, y \geq 2, z \geq 1$ is
 (A) 84 (B) 91 (C) 105 (D) 121
6. The solution of the recurrence relation $a_n = a_{n-1} + 2a_{n-2}$ with initial conditions $a_0 = 2, a_1 = 7$, is
 (A) $3(2)^n - (-1)^n$ (B) $3(2)^n + (-1)^n$
 (C) $-3(2)^n - (-1)^n$ (D) $-3(2)^n + (-1)^n$
7. Which among the following is not a subgroup of the set of Complex numbers under addition?
 (A) R , the set of all Real numbers.
 (B) Q^+ , the set of positive rational numbers.
 (C) Z , the set of all integers.
 (D) The set iR of purely imaginary numbers including 0
8. Minimum number n of integers to be selected from $S = \{1, 2, \dots, 9\}$ to guarantee that the difference of two of the n integers is 5 is
 (A) 3 (B) 4 (C) 6 (D) 9
9. Find the contrapositive the of statement “If it is a sunday, then I will wake up late”
 (A) If I am not waking up late, then it is a suniday
 (B) If I am not waking up late, then it is not a suniday
 (C) If it is not a sunday, then I will not wake up late.
 (D) It is not a sunday or I will wake up late
10. In the poset $(Z^+, |)$ (where Z^+ is the set of all positive integers and $|$ is the divides relation), which of the following are false?
 I. 3 and 9 is comparable
 II. 7 and 10 is comparable
 III. The poset $(Z^+, |)$ is a total order
 (A) I and III (B) II only (C) II and III (D) III only
11. Consider the following sequence of operations on an empty stack.
 push(22); push(43); pop(); push(55); push(12); s=pop();

Consider the following sequence of operations on an empty queue.

enqueue(32);enqueue(27); dequeue(); enqueue(38); enqueue(12); q=dequeue();

The value of s+q is _____

- (A) 44 (B) 54 (C) 39 (D) 70

12. The following postfix expression with single digit operands is evaluated using a stack:

8 2 2 ^ / 4 3 * + 5 1 * -

Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evaluated are:

- (A) 12,2 (B) 12,5 (C) 2,12 (D) 2,5

13. Construct a binary search tree by inserting 8, 6, 12, 3, 10, 9 one after another. To make the resulting tree as AVL tree which of the following is required?

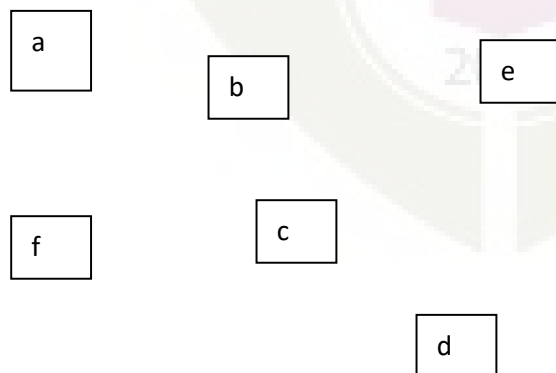
- (A) One right rotation only
 (B) One left rotation followed by two right rotations
 (C) One left rotation and one right rotation
 (D) The resulting tree itself is AVL

14. In a complete 4-ary tree, every internal node has exactly 4 children or no child. The number of leaves in such a tree with 6 internal nodes is:

- (A) 20 (B) 18 (C) 19 (D) 17

15. Consider the following graph with the following sequences

- I. a b c f d e
 II. a b e d f c
 III. a b f c d e
 IV. a f c b e d



Which are Depth First Traversals of the above graph?

- (A) I, II and IV only (B) I and IV only
 (C) II, III and IV only (D) I, III and IV only

16. Consider a hash table of size seven, with starting index zero, and a hash function $(2x + 5) \bmod 7$. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence 1, 4, 9, 6 is inserted into the table using closed hashing? Note that '_' denotes an empty location in the table.

- (A) 9, _, 1, 6, _, _, 4 (B) 1, _, 6, 9, _, _, 4
 (C) 4, _, 9, 6, _, _, 1 (D) 1, _, 9, 6, _, _, 4

17. Consider the following C program where `TreeNode` represents a node in a binary tree

```
struct TreeNode{
    struct TreeNode *leftChild;
    struct TreeNode *rightChild;
    int element;
};
int CountNodes(struct TreeNode *t)
{
    if((t==NULL)||((t->leftChild==NULL) && (t->rightChild==NULL)))
        return 0;
    else
    {
        return 1+CountNodes(t->leftChild)+CountNodes(t->rightChild)
    }
}
```

The value returned by `CountNodes` when a pointer to the root of a binary tree is passed as its argument is

- (A) number of nodes
 (B) number of leaf nodes
 (C) number of non leaf nodes
 (D) number of leaf nodes-number of non leaf nodes
18. How many distinct binary search trees can be created out of 6 distinct keys?
 (A) 7 (B) 36 (C) 140 (D) 132
19. Suppose a disk has 400 cylinders, numbered from 0 to 399. At some time the disk arm is at cylinder 58, and there is a queue of disk access requests for cylinder 66, 349, 201, 110, 38, 84, 226, 70, 86. If Shortest-Seek Time First (SSTF) is being used for scheduling the disk access, the request for cylinder 86 is serviced after servicing _____ number of

requests.

- (A) 1 (B) 2 (C) 3 (D) 4

20. If frame size is 4KB then a paging system with page table entry of 2 bytes can address _____ bytes of physical memory.

- (A) 2^{12} (B) 2^{16} (C) 2^{18} (D) 2^{28}

21. Calculate the internal fragmentation if page size is 4KB and process size is 103KB.

- (A) 3KB (B) 4KB (C) 1KB (D) 2KB

22. Which of the following scheduling policy is likely to improve interactiveness?

- (A) FCFS (B) Round Robin
(C) Shortest Process Next (D) Priority Based Scheduling

23. Consider the following program

Semaphore X=1, Y=0

```

Void A ( )
{
    While (1)
    {
        P(X);
        Print'1';
        V(Y);
    }
}

Void B ( )
{
    While (1)
    {
        P(Y);
        P(X);
        Print'0';
        V(X);
    }
}
    
```

The possible output of the program:

- (A) Any number of 0's followed by any number of 1's.
(B) Any number of 1's followed by any number of 0's.
(C) 0 followed by deadlock
(D) 1 followed by deadlock

24. In a system using single processor, a new process arrives at the rate of 12 processes per minute and each such process requires 5 seconds of service time. What is the percentage of CPU utilization?

- (A) 41.66 (B) 100.00 (C) 240.00 (D) 60.00

25. A system has two processes and three identical resources. Each process needs a maximum of two resources. This could cause

- (A) Deadlock is possible (B) Deadlock is not possible

- (C) Starvation may be present (D) Thrashing
26. Which of the following is true with regard to Round Robin scheduling technique?
- (A) Responds poorly to short process with small time quantum.
 (B) Works like SJF for larger time quantum
 (C) Does not use a prior knowledge of burst times of processes.
 (D) Ensure that the ready queue is always of the same size.
27. The size of the physical address space of a 32-bit processor is 2^W words. The capacity of cache memory is 2^N words. The size of each cache block is 2^K words. For a M-way set-associative cache memory, the length (in number of bits) of the tag field is
- (A) $W - N + \log_2 M$ (B) $W - N - \log_2 M$
 (C) $W - N - K - \log_2 M$ (D) $W - N - K + \log_2 M$
28. A 64-bit processor can support a maximum memory of 8 GB, where the memory is word-addressable (one word is of 64 bits). The size of the address bus of the processor is atleast _____ bits.
- (A) 30 (B) 31 (C) 32 (D) None
29. The stage delays in a 4-stage pipeline are 900, 450, 400 and 350 picoseconds. The first stage (with delay 900 picoseconds) is replaced with a functionally equivalent design involving two stages with respective delays 600 and 550 picoseconds. The throughput increase of the pipeline is _____ percent.
- (A) 38 (B) 30 (C) 58 (D) 50
30. Consider a direct mapped cache of size 256 Kilo words with block size 512 words. There are 6 bits in the tag. The number of bits in block (index) and word (offset) fields of physical address are is:
- (A) block (index) field = 6 bits, word (offset) field = 9 bits
 (B) block (index) field = 7 bits, word (offset) field = 8 bits
 (C) block (index) field = 9 bits, word (offset) field = 9 bits
 (D) block (index) field = 8 bits, word (offset) field = 8 bits
31. The memory unit of a computer has 1 Giga words of 64 bits each. The computer has instruction format, with 4 fields: an opcode field; a mode field to specify one of 12 addressing modes; a register address field to specify one of 48 registers; and a memory address field. If an instruction is 64 bits long, how large is the opcode field?
- (A) 34 bits (B) 24 bits (C) 20 bits (D) 14 bits
32. A computer has 64-bit instructions and 28-bit address. Suppose there are 252 two-address instructions. How many 1-address instructions can be formulated?

- (A) 2^{24} (B) 2^{26} (C) 2^{28} (D) 2^{30}

33. Determine the number of clock cycles required to process 200 tasks in a six-segment pipeline. (Assume there were no stalls), each segment takes 1 cycle.

- (A) 1200 cycles (B) 206 cycles (C) 207 cycles (D) 205 cycles

34. Match the following Lists:

- | | |
|--------------------------|-------------------------------|
| P. DMA | 1. Priority Interrupt |
| Q. Processor status Word | 2. I/O Transfer |
| R. Daisy chaining | 3. CPU |
| S. Handshaking | 4. Asynchronous Data Transfer |

- (A) P-1, Q-3, R-4, S-2 (B) P-2, Q-3, R-1, S-4
 (C) P-2, Q-1, R-3, S-4 (D) P-4, Q-3, R-1, S-2

35. Let E1, E2 and E3 be three entities in an E/R diagram with simple single-valued attributes. R1 and R2 are two relationships between E1 and E2, where R1 is one-to-many, R2 is many-to-many. R3 is another relationship between E2 and E3 which is many-to-many. R1, R2 and R3 do not have any attributes of their own. What is the minimum number of tables required to represent this situation in the relational model?

- (A) 3 (B) 4 (C) 5 (D) 6

36. Identify the minimal key for relational scheme R(U, V, W, X, Y, Z) with functional dependencies $F = \{U \rightarrow V, V \rightarrow W, W \rightarrow X, VX \rightarrow Z\}$

- (A) UV (B) UW (C) UX (D) UY

37. It is given that: "Every student need to register one course and each course registered by many students", what is the cardinality of the relation say "Register" from the "Student" entity to the "Course" entity in the ER diagram to implement the given requirement.

- (A) M:1 relationship (B) M:N relationship
 (C) 1:1 relationship (D) option (B) or(C)

38. Consider the relation branch(branch_name, assets, branch_city)

SELECT DISTINCT T.branch_name FROM branch T, branch S WHERE T.assets > L.assets AND S.branch_city = "TVM" .

Finds the names of

- (A) All branches that have greater assets than all branches located in TVM.
 (B) All branches that have greater assets than some branch located in TVM.
 (C) The branch that has the greatest asset in TVM.
 (D) Any branch that has greater asset than any branch located in TVM.

39. Consider the following relation instance, where “A” is primary Key.

A1	A2	A3	A4
1	1	1	Null
5	2	5	1
9	5	13	5
13	13	9	15

Which one of the following can be a foreign key that refers to the same relation?

- (A) A2 (B) A3 (C) A4 (D) ALL

40. A relation R(ABC) is having the tuples(1,2,1),(1,2,2),(1,3,1) and (2,3,2). Which of the following functional dependencies holds well?

- (A) $A \rightarrow BC$ (B) $AC \rightarrow B$ (C) $AB \rightarrow C$ (D) $BC \rightarrow A$

41. Consider a relation R with attributes A, B, C, D and E and functional dependencies $A \rightarrow BC$, $BC \rightarrow E$, $E \rightarrow DA$. What is the highest normal form that the relation satisfies?

- (A) BCNF (B) 3 NF (C) 2 NF (D) 1 NF

42. For the given schedule S, find out the conflict equivalent schedule.

S : r1(x); r2(Z) ; r3(X); r1(Z); r2(Y); r3(Y); W1(X); W2(Z); W3(Y); W2(Y)

- (A) $T1 \rightarrow T2 \rightarrow T3$ (B) $T2 \rightarrow T1 \rightarrow T3$
 (C) $T3 \rightarrow T1 \rightarrow T2$ (D) Not conflict serializable

43. Which of the following strings is in the language defined by the grammar:

$S \rightarrow aX$

$X \rightarrow aX \mid bX \mid b$

- (A) aaaba (B) babab (C) aaaaa (D) ababb

44. Consider the regular expression $(x+y)^*xyx(x+y)^*$ where $\Sigma = (x,y)$. If L is the language represented by this regular expression, then what will be the minimum number of states in a DFA recognizing L ?

- (A) 2 (B) 3 (C) 4 (D) 5

45. Which of the following cannot handle the same set of languages?

- (A) Deterministic Finite Automata and Non-Deterministic Finite Automata
 (B) Deterministic Push Down Automata and Non-Deterministic Push Down Automata
 (C) All of these
 (D) None of these

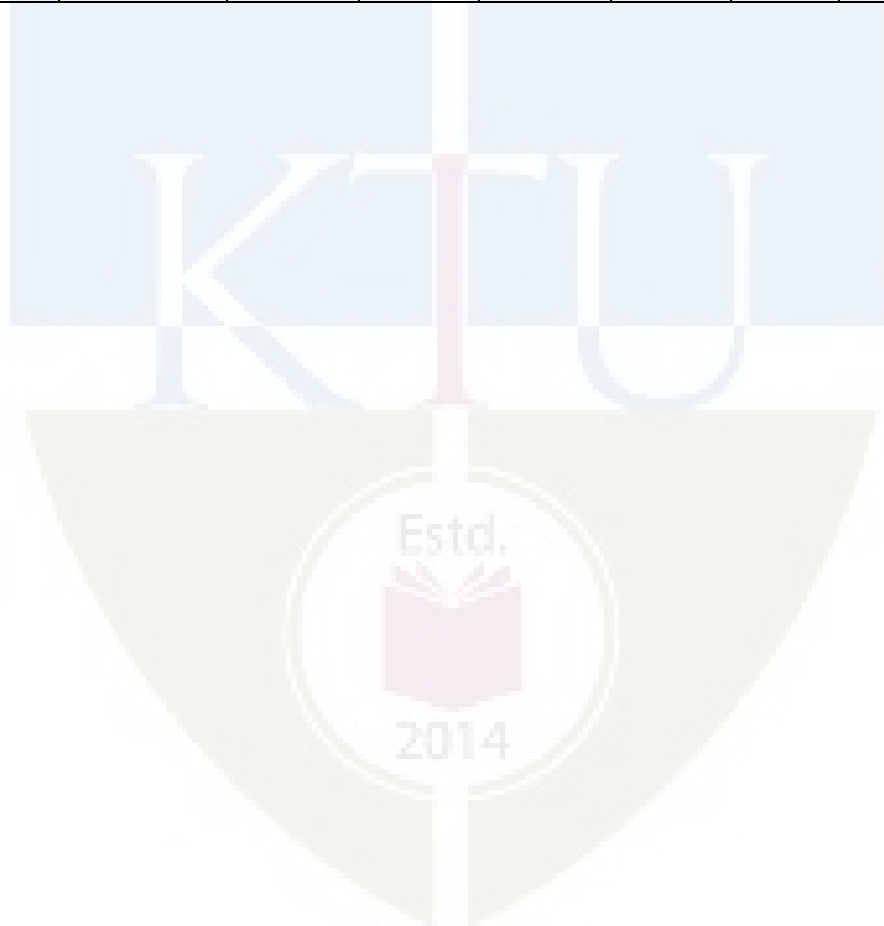
46. Consider L be a context-free language and M be a non-context-free language. Which among the following is TRUE?

- (I) L will definitely pass the pumping lemma test for CFLs.
 (II) M will definitely pass the pumping lemma test for CFLs.
 (III) L will not definitely pass the pumping lemma test for CFLs.
 (IV) M will not definitely pass the pumping lemma test for CFLs.
 (V) L may or maynot pass the pumping lemma test for CFLs.
 (VI) M may or maynot pass the pumping lemma test for CFLs.
 (A) I, II (B) II, V (C) I, VI (D) IV, V
47. Which of the following problem(s) is/are decidable?
 (I) Whether a CFG is empty or not.
 (II) Whether a CFG generates all possible strings.
 (III) Whether the language generated by a Turing Machine is regular.
 (IV) Whether the language generated by DFA and NFA are same.
 (A) I and II (B) II and III (C) II and IV (D) I and IV
48. Which of the following is/are TRUE?
 (I) Regular languages are closed under complementation.
 (II) Recursive languages are closed under complementation.
 (III) Context free languages are closed under complementation.
 (IV) Context free languages are not closed under complementation.
 (A) I, II and III (B) I, II and IV (C) II and III (D) III only
49. Which of the following regular expressions defined over the alphabet $\Sigma = \{0,1\}$ defines the language of all strings of length l where l is a multiple of 3?
 (A) $(0 + 1 + 00 + 11 + 000 + 111)^*$ (B) $(000 + 111)^*$
 (C) $((0 + 1)(0 + 1)(0 + 1))^*$ (D) $((000 + 01 + 1)(111 + 10 + 0))^*$
50. Determine the minimum number of states of a DFA that recognizes the language over the alphabet $\{a,b\}$ consisting of all the strings that contain at least three a's and at least four b's.
 (A) 6 (B) 12 (C) 15 (D) 20

ANSWER KEY:-

QNo	Ans. Key	QNo	Ans. Key	QNo	Ans. Key	QNo	Ans. Key	QNo	Ans. Key
1	(C)	11	(C)	21	(C)	31	(B)	41	(A)

2	(A)	12	(A)	22	(B)	32	(D)	42	(D)
3	(B)	13	(A)	23	(D)	33	(D)	43	(D)
4	(C)	14	(C)	24	(B)	34	(B)	44	(C)
5	(B)	15	(A)	25	(B)	35	(C)	45	(B)
6	(A)	16	(D)	26	(C)	36	(D)	46	(C)
7	(B)	17	(C)	27	(A)	37	(A)	47	(D)
8	(C)	18	(D)	28	(A)	38	(B)	48	(B)
9	(B)	19	(C)	29	(D)	39	(B)	49	(C)
10	(C)	20	(D)	30	(C)	40	(D)	50	(D)



CSL332	NETWORKING LAB	CATEGORY	L	T	P	Credit	Year of Introduction
		PCC	0	0	3		

Preamble:

The course enables the learners to get hands-on experience in network programming using Linux System calls and network monitoring tools. It covers implementation of network protocols and algorithms, configuration of network services and familiarization of network simulators. This helps the learners to develop, implement protocols and evaluate its performance for real world networks.

Prerequisite: Sound knowledge in Programming in C, Data Structures and Computer Networks

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Use network related commands and configuration files in Linux Operating System. (Cognitive Knowledge Level: Understand).
CO2	Develop network application programs and protocols. (Cognitive Knowledge Level: Apply)
CO3	Analyze network traffic using network monitoring tools. (Cognitive Knowledge Level: Apply)
CO4	Design and setup a network and configure different network protocols. (Cognitive Knowledge Level: Apply)
CO5	Develop simulation of fundamental network concepts using a network simulator. (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												

C04												
C05												

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and teamwork
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Test (Internal Exam) Marks in percentage	End Semester Examination Marks in percentage
Remember	20	20
Understand	20	20
Apply	60	60
Analyze		
Evaluate		
Create		

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 15 marks
Continuous Evaluation in Lab	: 30 marks
Continuous Assessment Test	: 15 marks
Viva voce	: 15 marks

Internal Examination Pattern:

The Internal examination shall be conducted for 100 marks, which will be converted to out of 15, while calculating Internal Evaluation marks. The marks will be distributed as, Algorithm - 30 marks, Program - 20 marks, Output - 20 marks and Viva - 30 marks.

End Semester Examination Pattern:

The End Semester Examination will be conducted for a total of 75 marks and shall be distributed as, Algorithm - 30 marks, Program - 20 marks, Output - 20 marks and Viva- 30 marks.

Operating System to Use in Lab	: Linux
Compiler/Software to Use in Lab	: gcc, NS2
Programming Language to Use in Lab	: Ansi C

Fair Lab Record:

All the students attending the Networking Lab should have a Fair Record. Every experiment conducted in the lab should be noted in the fair record. For every experiment, in the fair record, the right hand page should contain experiment heading, experiment number, date of experiment, aim of the experiment, procedure/algorithm followed, other such details of the experiment and final result. The left hand page should contain a print out of the respective code with sample input and corresponding output obtained. All the experiments noted in the fair record should be verified by the faculty regularly. The fair record, properly certified by the faculty, should be produced during the time of End Semester Examination for the verification by the examiners.

Syllabus

*Mandatory

(Note: At least one program from each topic in the syllabus should be completed in the Lab)

1. Getting started with the basics of network configuration files and networking commands in Linux.*
2. To familiarize and understand the use and functioning of system calls used for network programming in Linux.*
3. Implement client-server communication using socket programming and TCP as transport layer protocol*
4. Implement client-server communication using socket programming and UDP as transport layer protocol*
5. Simulate sliding window flow control protocols.* (Stop and Wait, Go back N, Selective Repeat ARQ protocols)
6. Implement and simulate algorithm for Distance Vector Routing protocol or Link State Routing protocol.*
7. Implement Simple Mail Transfer Protocol.
8. Implement File Transfer Protocol.*
9. Implement congestion control using a leaky bucket algorithm.*
10. Understanding the Wireshark tool.*
11. Design and configure a network with multiple subnets with wired and wireless LANs using required network devices. Configure commonly used services in the network.*
12. Study of NS2 simulator*

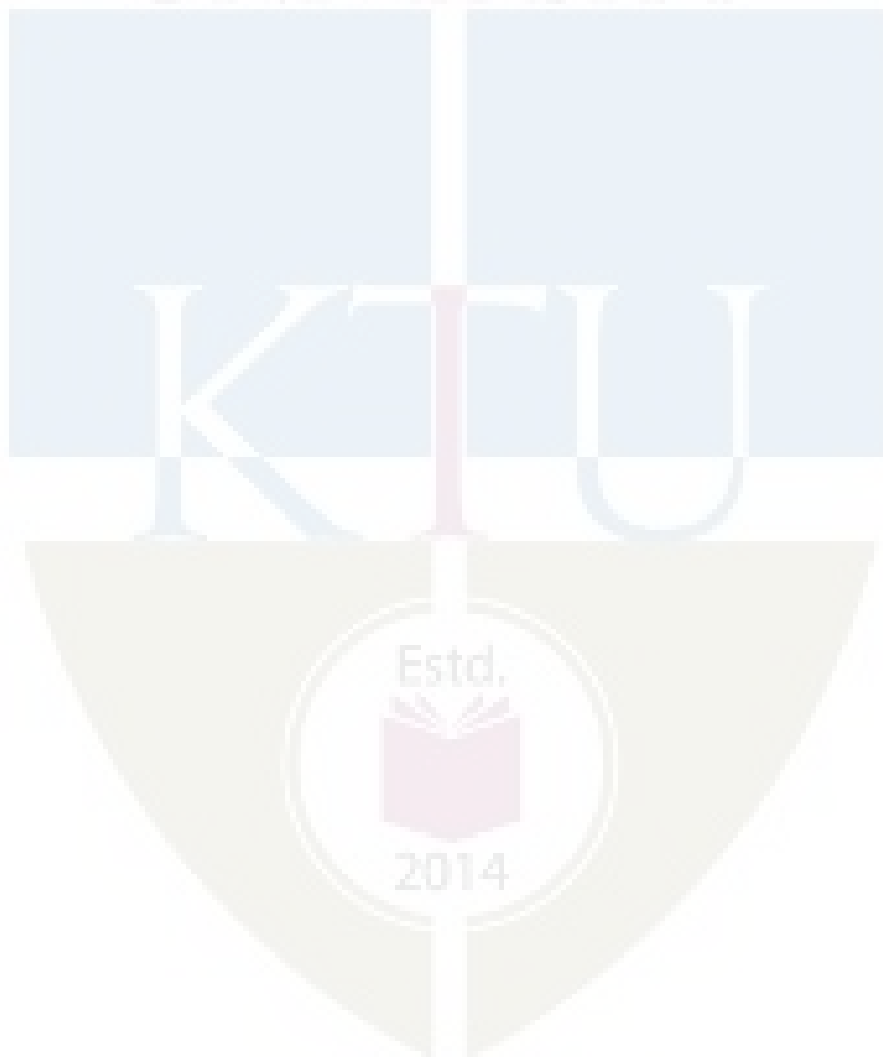
Networking Lab-Practice Questions

1.
 - a) View the configuration, including addresses of your computers network interfaces.
 - b) Test the network connectivity between your computer and several other computers.
 - c) View the active TCP connections in the computer after visiting a website.
 - d) Find the hardware/MAC address of another computer in the network using ARP.
2. Write the system calls used for creating sockets and transferring data between two nodes.
3.
 - a) Implement a multi-user chat server using TCP as transport layer protocol.
 - b) Implement a simple web proxy server that accepts HTTP requests and forwarding to remote servers and returning data to the client using TCP

4. Implement a Concurrent Time Server application using UDP to execute the program at a remote server. Client sends a time request to the server, server sends its system time back to the client. Client displays the result.
5.
 - a) Implement Stop-and-Wait ARQ flow control protocol.
 - b) Implement Go-Back--N ARQ flow control protocol.
 - c) Implement Selective Repeat ARQ flow control protocol.
6. Implement Distance Vector Routing algorithm or Link State Routing algorithm..
7. Implement Simple Mail Transfer Protocol.
8. Develop a concurrent file server which will provide the file requested by a client if it exists. If not, the server sends appropriate message to the client. Server should also send its process ID (PID) to clients for display along with the file or the message.
9. Implement leaky bucket algorithm for congestion control.
10.
 - a) Using Wireshark, Capture packets transferred while browsing a selected website. Investigate the protocols used in each packet, the values of the header fields and the size of the packet.
 - b) Using Wireshark, observe three way handshaking connection establishment, three way handshaking connection termination and Data transfer in client server communication using TCP.
 - c) Explore at least the following features of Wireshark: filters, Flow graphs (TCP), statistics, and protocol hierarchies.
11. Design and configure a network (wired and wireless LANs) with multiple subnets using required network devices. Configure at least three of the following services in the network- TELNET, SSH, FTP server, Web server, File server, DHCP server and DNS server.
12.
 - a) The network consists of TCP source node (n_0) and destination node (n_1) over an area size of 500m x 500m. Node (n_0) uses Agent/TCP/Reno as the sending TCP agent and FTP traffic source. Node (n_1) is the receiver of FTP transfers, and it uses Agent/TCP sink as its TCP-agent for the connection establishment. Run the simulation for 150 seconds and show the TCP window size in two static nodes scenario with any dynamic routing protocol. Run the script and analyze the output graph for the given scenario.
 - b) Simulate the transmission of ping messages over a star network topology consisting of 'n' nodes and find the number of packets dropped due to congestion using NS2simulator.
 - c) Simulate Link State Protocol or Distance Vector Routing protocol in NS2.

Reference Books:

1. W. Richard Stevens, Bill Fenner, Andy Rudoff, UNIX Network Programming: Volume 1, The Sockets Networking API, 3rd Edition, Pearson, 2015
2. Lisa Bock, Learn Wireshark: Confidently navigate the Wireshark interface and solve real-world networking problems, Packt Publishing, 2019
3. Teerawat Issariyakul, Ekram Hossain, Introduction to Network Simulator NS2, 2nd Edition, Springer, 2019



CSD 334	MINI PROJECT	Category	L	T	P	Credit	Year of Introduction
		PCC	0	0	3	2	2019

Preamble:

The objective of this course is to apply the fundamental concepts of Software Engineering principles for the effective development of an application/research project. This course helps the learners to practice the different steps to be followed in the software development process such as literature review and problem identification, preparation of Software Requirement Specification & Software Design Document (SDD), testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

Prerequisite:

A sound knowledge in any programming language and fundamental concepts of Software Engineering.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CO2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CO3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CO4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CO5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Abstract POs defined by National Board of Accreditation

PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	

Continuous Internal Evaluation Pattern:

Attendance	10 marks
Project Guide	15 marks
Project Report	10 marks

Evaluation by the Committee (will be evaluating the level of completion and demonstration of functionality/specifications, presentation, oral examination, work knowledge and involvement) : **40 marks**

Student Groups with 3 or 4 members should identify a topic of interest in consultation with a Faculty/Advisor. Review the literature and gather information pertaining to the chosen topic. State the objectives and develop a methodology to achieve the objectives. Carryout the design/fabrication or develop codes/programs to achieve the objectives by strictly following steps specified in the teaching plan. Innovative design concepts, performance, scalability, reliability considerations, aesthetics/ergonomic, user experience and security aspects taken care of in the project shall be given due weight.

The progress of the mini project is evaluated based on a minimum of two reviews. The review committee may be constituted by the Head of the Department comprising HoD or a senior faculty member, Mini Project coordinator and project guide. The internal evaluation shall be made based on the progress/outcome of the project, reports and a viva-voce examination, conducted internally by a 3-member committee. A project report is required at the end of the semester. The project has to be demonstrated for its full design specifications.

End Semester Examination Pattern:

The marks will be distributed as

Presentation	: 30 marks
Demo	: 20 marks
Viva	: 25 marks.
Total	: 75 marks.

COMPUTER SCIENCE AND ENGINEERING TEACHING PLAN

Students are expected to follow the following steps.

1. Review of Literature and Identification of a problem
2. Create an abstract with a problem statement, solution approach, technology stack, team, etc. and get department approval. Register Online course/ Collect study materials.
3. Create Software Requirements Specification (SRS Document)
4. Create Software Design Document (SDD). This may include designs like,
 - a. System Architecture Design
 - b. Application Architecture Design
 - c. GUI Design (Mockups)
 - d. API Design
 - e. Database Design
 - f. Technology Stack
5. Create Test Plan, Test Scenarios and Test Cases (Test Case Document) & Traceability Matrix
6. Create a Project Plan (with Modules, Tasks, Resources, Time schedule) [May use any project management tool or excel for this] – Choose and follow agile or waterfall models.
7. Development
 - a. Set coding standards
 - b. Environment Setup
 - c. Source Code Control Setup (Like Subversion(SVN), Git)
 - d. Development
 - e. Unit Testing
 - f. Integration Testing
 - g. Testing /Quality Assurance(QA)
 - i. Functional Testing
 - ii. Load Testing
 - iii. Report Bugs
 - h. Resolve Bugs & Retest

8. Deployment (of software from local development environment to a production environment)
9. Test Run & Get Results
10. Prepare Project Report

Guidelines for the Report preparation

A bonafide report on the mini project shall be submitted within one week after the final presentation. Minimum number of pages should be 40.

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman 18, Bold; Heading 2 – Times New Roman 16, Bold; Heading 3 – Times New Roman 14, Bold; Body- Times New Roman 12, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.
- Figures & Tables – Ensure that all Figures and Tables are suitably numbered and given proper names/headings. Write figure title under the figure and table title above the table.
- **Suggestive order of documentation:**
 - i. Top Cover
 - ii. Title page
 - iii. Certification page
 - iv. Acknowledgement
 - v. Abstract
 - vi. Table of Contents
 - vii. List of Figures and Tables
 - viii. Chapters
 - ix. Appendices, if any
 - x. References/Bibliography

APJ ABDUL KALAM
TECHNOLOGICAL
UNIVERSITY

SEMESTER VI

PROGRAM ELECTIVE I



CST 312	FOUNDATIONS OF MACHINE LEARNING	Category	L	T	P	Credit	Year of Introduction
		PEC	2	1	0	3	2019

Preamble:

This course enables the learners to understand the mathematical foundations of Machine Learning concepts. This course covers Linear Algebra, Probability and Distributions. Concepts in this course help the learners to identify the inherent assumptions & limitations of the current methodologies and develop new Machine Learning solutions.

Prerequisite: A sound background in higher secondary school Mathematics.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Illustrate operations and applications of linear equations, matrix algebra, vector spaces, eigen values & eigenvectors (Cognitive Knowledge Level: Apply)
CO 2	Illustrate the concepts of orthogonality & diagonalization. (Cognitive Knowledge Level: Apply)
CO 3	Solve computational problems using probability and random variables. (Cognitive Knowledge Level: Apply)
CO 4	Identify an appropriate probability distribution for a given discrete or continuous random variable and use its properties. (Cognitive Knowledge Level: Apply)
CO 5	Illustrate moment generating function, law of large numbers and central limit theorems (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	✓	✓	✓	✓								✓
CO 2	✓	✓	✓	✓								✓
CO 3	✓	✓	✓	✓								✓
CO 4	✓	✓	✓	✓								✓

CO 5	✓	✓	✓	✓								✓
------	---	---	---	---	--	--	--	--	--	--	--	---

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	30%	30%	30%
Understand	30%	30%	30%
Apply	40%	40%	40%
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer anyone. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

Module 1 (LINEAR ALGEBRA)

Systems of Linear Equations – Matrices, Solving Systems of Linear Equations. Vector Spaces - Linear Independence, Basis and Rank, Linear Mappings.

Module 2 (LINEAR ALGEBRA)

Norms - Inner Products, Lengths and Distances, Angles and Orthogonality. Orthonormal Basis, Orthogonal Complement, Orthogonal Projections. Matrix Decompositions - Eigenvalues and Eigenvectors, Eigen decomposition and Diagonalization.

Module 3 (PROBABILITY AND DISTRIBUTIONS)

Probability Space - Sample Spaces, Probability Measures, Computing Probabilities, Conditional Probability, Baye's Rule, Independence. Random Variables - Discrete Random Variables (Bernoulli Random Variables, Binomial Distribution, Geometric and Poisson Distribution, Continuous Random Variables (Exponential Density, Gamma Density, Normal Distribution, Beta Density)

Module 4 (RANDOM VARIABLES)

Functions of a Random Variable. Joint Distributions - Independent Random Variables, Conditional Distributions, Functions of Jointly Distributed Random Variables.

Expected Values - Expected Value of a Random Variable, Expectations of Functions of Random Variables, Expectations of Linear Combinations of Random Variables, Variance and Standard Deviation, Covariance and Correlation, Conditional Expectation

Module 5 (LIMIT THEOREMS)

Moment-Generating Function. Limit Theorems(Proof not expected) - Law of Large Numbers, Convergence in Distribution and the Central Limit Theorem. Distributions derived from the Normal Distribution - Chi-square, t, and F Distributions, Sample Mean and the Sample Variance.

Text book:

1. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, Mathematics for Machine Learning, Cambridge University Press (freely available at [https://mml – book.github.io](https://mml-book.github.io))
2. John A. Rice, Mathematical Statistics and Data Analysis, University of California, Berkeley, Third edition, published by Cengage.

Reference books:

1. Gilbert Strang, Linear Algebra and Its Applications, 4th Edition,
2. Axler, Sheldon, Linear Algebra Done Right, 2015 Springer
3. Stephen Boyd and Lieven Vandenberghe, Introduction to Applied Linear Algebra, 2018 published by Cambridge University Press

Sample Course Level Assessment Questions**Course Outcome 1 (CO1):**

1. Find the set S of all solutions in x of the following inhomogeneous linear systems $Ax = b$, where A and b are defined as follows:

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

2. Determine the inverses of the following matrix if possible

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

3. Are the following sets of vectors linearly independent?

$$x_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

4. A set of n linearly independent vectors in R^n forms a basis. Does the set of vectors $(2, 4, -3)$, $(0, 1, 1)$, $(0, 1, -1)$ form a basis for R^3 ? Explain your reasons.

Course Outcome 2 (CO2):

1. Determine which of the following sets are orthogonal sets.

$$\left\{ \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} -1/2 \\ -2 \\ 7/2 \end{bmatrix} \right\} \quad \left\{ \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 3 \\ 0 \\ -3 \end{bmatrix} \right\} \quad \left\{ \begin{bmatrix} 3 \\ -2 \\ 1 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \\ -3 \\ 4 \end{bmatrix}, \begin{bmatrix} 3 \\ 8 \\ 7 \\ 0 \end{bmatrix} \right\}$$

2. Find the characteristic equation, eigenvalues, and eigenspaces corresponding to each eigenvalue of the following matrix.

$$\begin{bmatrix} 2 & 0 & 4 \\ 0 & 3 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

3. Diagonalize the following matrix, if possible

$$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 3 \end{bmatrix}$$

Course Outcome 2 (CO3):

1. Let J and T be independent events, where $P(J)=0.4$ and $P(T)=0.7$.
 - i. Find $P(J \cap T)$
 - ii. Find $P(J \cup T)$
 - iii. Find $P(J \cap T')$
2. Let A and B be events such that $P(A)=0.45$, $P(B)=0.35$ and $P(A \cup B)=0.5$. Find $P(A/B)$.
3. A random variable \mathbf{R} has the probability distribution as shown in the following table:

r	1	2	3	4	5
P(R=r)	0.2	a	b	0.25	0.15

- i. Given that $E(R)=2.85$, find a and b .
 - ii. Find $P(R>2)$.
4. A biased coin (with probability of obtaining a head equal to $p > 0$) is tossed repeatedly and independently until the first head is observed. Compute the probability that the first head appears at an even numbered toss.
 5. Two players A and B are competing at a quiz game involving a series of questions. On any individual question, the probabilities that A and B give the correct answer are p and q respectively, for all questions, with outcomes for different questions being independent. The game finishes when a player wins by answering a question correctly. Compute the probability that A wins if
 - i. A answers the first question,
 - ii. B answers the first question.
 6. A coin for which $P(\text{heads}) = p$ is tossed until two successive tails are obtained. Find the probability that the experiment is completed on the n^{th} toss.

Course Outcome- 3 (CO4):

1. An urn contains p black balls, q white balls, and r red balls; and n balls are chosen without replacement.
 - a. Find the joint distribution of the numbers of black, white, and red balls in the sample.
 - b. Find the joint distribution of the numbers of black and white balls in the sample.
 - c. Find the marginal distribution of the number of white balls in the sample.
2. Suppose that two components have independent exponentially distributed lifetimes, T_1 and T_2 , with parameters α and β , respectively. Find (a) $P(T_1 > T_2)$ and (b) $P(T_1 > 2 T_2)$.
3. Let Z_1 and Z_2 be independent random variables each having the standard normal distribution. Define the random variables X and Y by $X = Z_1 + 3Z_2$ and $Y = Z_1 + Z_2$. Argue that the joint distribution of (X, Y) is a bivariate normal distribution. What are the parameters of this distribution?

4. Given a continuous random variable x , with cumulative distribution function $F_x(x)$, show that the random variable $y = F_x(x)$ is uniformly distributed.
5. You roll a fair dice twice. Let the random variable X be the product of the outcomes of the two rolls. What is the probability mass function of X ? What are the expected values and the standard deviation of X ?
6. Let X be a continuous random variable with the density function $f(x) = 2x$, $0 \leq x \leq 1$
 - a. Find $E(X)$.
 - b. Find $E(X^2)$ and $Var(X)$.

Course Outcome 5 (CO5):

1. Find the moment-generating function of a Bernoulli random variable, and use it to find the mean, variance, and third moment.
2. Use moment-generating functions to show that if X and Y are independent, then $Var(aX + bY) = a^2Var(X) + b^2Var(Y)$.
3. Suppose that you bet Rs 5 on each of a sequence of 50 independent fair games. Use the central limit theorem to approximate the probability that you will lose more than Rs 75.
4. Suppose that the number of insurance claims, N , filed in a year is Poisson distributed with $E(N) = 10,000$. Use the normal approximation to the Poisson to approximate $P(N > 10,200)$.

Model Question paper

QP Code :

Total Pages: 4

Reg No.: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION (ELECTIVE), MONTH and YEAR

Course Code: CST 312**Course Name: FOUNDATIONS OF MACHINE LEARNING**

Max. Marks: 100

Duration: 3 Hours

PART A*Answer all questions, each carries 3 marks.*

- 1 Show that with the usual operation of scalar multiplication but with addition on reals given by $x \# y = 2(x + y)$ is not a vector space.
- 2 Are the following vectors linearly independent? Justify your answer.

$$x_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

- 3 Find the eigenvalues of the following matrix in terms of k. Can you find an eigenvector corresponding to each of the eigenvalues?

$$\begin{bmatrix} 1 & k \\ 2 & 1 \end{bmatrix}$$

- 4 Find a unit vector in \mathbf{R}^2 that is orthogonal to $(-1, 2)$.
- 5 The first three digits of a telephone number are 452. If all the sequences of the remaining four digits are equally likely, what is the probability that a randomly selected telephone number contains seven distinct digits?

- 6 Show that if two events A and B are independent, then A and B' are independent.
- 7 Prove that X and Y are independent if and only if $f_{X|Y}(x|y) = f_X(x)$ for all x and y .
- 8 If X is a discrete uniform random variable, i.e., $P(X = k) = 1/n$ for $k = 1, 2, \dots, n$, find $E(X)$ and $Var(X)$.
- 9 Compare the Poisson cdf and the normal approximation for (a) $\lambda = 10$, (b) $\lambda = 20$, and (c) $\lambda = 40$.
- 10 State law of large numbers. 10 x 3 = 30

PART B

Answer any one Question from each module. Each question carries 14 Marks

- 11 a) Find all solutions to the system of linear equations (8)

$$\begin{aligned} -4x + 5z &= -2 \\ -3x - 3y + 5z &= 3 \\ -x + 2y + 2z &= -1 \end{aligned}$$

- b) Consider the transformation $T(x, y) = (x + y, x + 2y, 2x + 3y)$. Obtain $\ker T$ and use this to calculate the nullity. Also find the transformation matrix for T . (6)

OR

- 12 a) Consider the following linear mapping (8)

$$\begin{aligned} \Phi : \mathbb{R}^3 &\rightarrow \mathbb{R}^4 \\ \Phi \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right) &= \begin{bmatrix} 3x_1 + 2x_2 + x_3 \\ x_1 + x_2 + x_3 \\ x_1 - 3x_2 \\ 2x_1 + 3x_2 + x_3 \end{bmatrix} \end{aligned}$$

- i. Find the transformation matrix T .
- ii. Determine $\text{rank}(T)$.

iii. Compute the kernel and image of the mapping and find their dimension

b) Prove that all vectors orthogonal to $[2, -3, 1]^T$ forms a subspace W of R^3 . What is $\dim(W)$ and why? (6)

13 a) Find an orthonormal basis of R^3 consisting of eigenvectors for the following matrix (8)

$$\begin{bmatrix} 1 & 0 & -2 \\ 0 & 5 & 0 \\ -2 & 0 & 4 \end{bmatrix}$$

b) Find a 3×3 orthogonal matrix S and a 3×3 diagonal matrix D such that $A = SDS^T$ (6)

OR

14 a) Find an orthogonal basis for the subspace of R^4 spanned by $\{w_1 = (1, 1, 3, 2), w_2 = (1, -2, 0, -1), w_3 = (0, 2, 1, 2)\}$. (8)

b) Find the characteristic equation, eigenvalues, and eigenspaces corresponding to each eigenvalue of the following matrix (6)

$$\begin{bmatrix} 2 & 0 & 4 \\ 0 & 3 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

15 a) Three players play 10 independent rounds of a game, and each player has probability $1/3$ of winning each round. Find the joint distribution of the numbers of games won by each of the three players. (7)

b) An experiment consists of throwing a fair coin four times. Find the probability mass function and the cumulative distribution function of the following random variables: (7)

- i. the number of heads before the first tail
- ii. the number of heads following the first tail
- iii. the number of heads minus the number of tails
- iv. the number of tails times the number of heads.

OR

- 16 a) A factory runs three shifts. On a given day, 1% of the items produced by the first shift are defective, 2% of the second shift's items are defective, and 5% of the third shift's items are defective. If the shifts all have the same productivity, what percentage of the items produced in a day are defective? If an item is defective, what is the probability that it was produced by the third shift? (8)
- b) Show that if A and B are two independent events, then $P(A \cup B) = P(A) + P(B) - P(A)P(B)$ (6)
- 17 a) Find the joint density of $X + Y$ and X/Y , where X and Y are independent exponential random variables with parameter λ . Show that $X + Y$ and X/Y are independent. (8)
- b) Let X be a discrete random variable that takes on values 0, 1, 2 with probabilities $1/2, 3/8, 1/8$, respectively. (6)
- i. Find $E(X)$ and $\text{Var}(X)$.
 - ii. Let $Y = X^2$. Find the probability mass function of Y and use it to find $E(Y)$.
- 18 a) A random square has a side length that is a uniform $[0, 1]$ random variable. Find the expected area of the square. (7)
- b) Let X be a continuous random variable with probability density function on $0 \leq x \leq 1$ defined by $f(x) = 3x^2$. Find the pdf of $Y = X^2$. (7)
- 19 a) Using the fact that the mean of the chi-squared distribution is $(n-1)\sigma^2$, prove that $E(S^2) = \sigma^2$. (7)
- b) i. Random samples of size 36 are taken from an infinite population whose mean is 80 and standard deviation is 18. Find the mean and standard error of the (7)

sampling distribution.

ii. Why is the Central Limit Theorem so important to statistical analysis?

OR

20 a) A six-sided die is rolled 100 times. Using the normal approximation, find the probability that the face showing a six turns up between 15 and 20 times. Find the probability that the sum of the face values of the 100 trials is less than 300. **(8)**

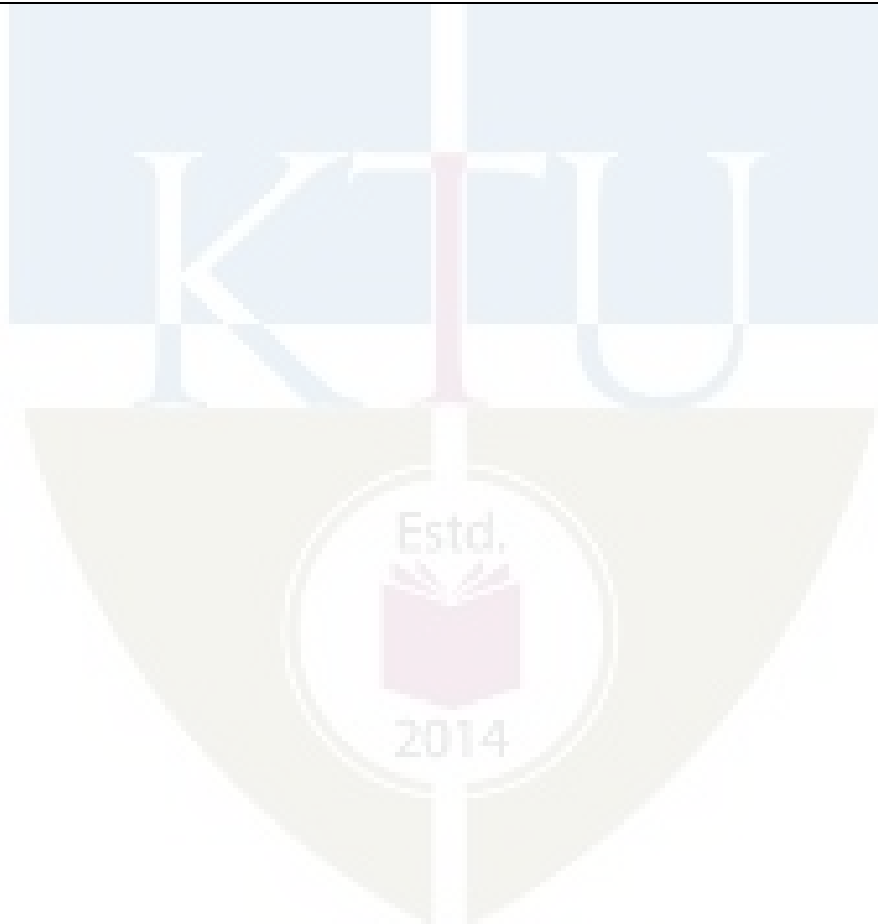
b) Determine an interval (a, b) such that $P[a \leq t \leq b] = 0.80$, and that 10% of the area is on each side of a and b , assuming that the sample is of size 21. **(6)**



Teaching Plan		
No	Topic	No. of Lectures (35)
1	Module-1 (LINEAR ALGEBRA) TB-1(Ch 2,3,4) (6 hours)	
1.1	Systems of Linear Equations – Matrices, Solving Systems of Linear Equations.	1 hour
1.2	Vector Spaces, sub space	1 hour
1.3	Linear Independence,	1 hour
1.4	Basis and Rank	1 hour
1.5.	Linear Mappings- Kernel, Range	1 hour
1.6.	Linear Mappings- Rank, Nullity	
2	Module-2 (LINEAR ALGEBRA) (6 hours)	
2.1.	Norms, Inner Products, Lengths and Distances, Angles and Orthogonality,	1 hour
2.2	Orthonormal Basis, Orthogonal Complement,	1 hour
2.3	Orthogonal Projections	1 hour
2.4.	Eigenvalues and Eigenvectors	1 hour
2.5.	Eigen decomposition	1 hour
2.6.	Eigen Diagonalization	1 hour
3.	Module-3 (PROBABILITY AND DISTRIBUTIONS) TB-2(Ch 1,2) (9 hours)	

3.1	Sample Spaces, Probability Measures, Computing Probabilities	1 hour
3.2	Conditional Probability,	1 hour
3.3	Baye's Rule	1 hour
3.4	Independence of events	1 hour
3.5	Discrete Random Variables -Bernoulli Random Variables, Binomial Distribution	1 hour
3.6	Discrete Random Variables -Geometric Distribution	1 hour
3.7	Discrete Random Variables -Poisson Distribution	1 hour
3.8	Continuous Random Variables - Exponential Density, Gamma Density,	1 hour
3.9	Continuous Random Variables - Normal Distribution, Beta Density	1 hour
4.	Module-4 (RANDOM VARIABLES) TB-2 (Ch 3, 4, 5, 6) (9 hours)	
4.1	Functions of a Random Variable	1 hour
4.2	Joint Distributions - Independent Random Variables	1 hour
4.3	Conditional Distributions	1 hour
4.4	Functions of Jointly Distributed Random Variables	1 hour
4.5	Expected Value of a Random Variable,	1 hour
4.6	Expectations of Functions of Random Variables,	1 hour
4.7	Expectations of Linear Combinations of Random Variables	1 hour
4.6	Variance and Standard Deviation	1 hour
4.9	Covariance and Correlation	1 hour

5	Module-5 (LIMIT THEOREMS) (6 hours)	
5.1	Conditional Expectation,	1 hour
5.2	Moment-Generating Function	1 hour
5.3	Limit Theorems(Proof not expected) - Law of Large Numbers,	1 hour
5.4	Convergence in Distribution and the Central Limit Theorem.	1 hour
5.5	Distributions derived from the Normal Distribution - Chi-square and, and F Distributions,	1 hour
5.6	Distributions derived from the Normal Distribution - Sample Mean and the Sample Variance.	1 hour



CST 322	DATA ANALYTICS	Category	L	T	P	Credits	Year of Introduction
		PEC	2	1	0	3	2019

Preamble:

This course helps the learner to understand the basic concepts of data analytics. This course covers mathematics for data analytics, predictive and descriptive analytics of data, Big data and its applications, techniques for managing big data and data analysis & visualization using R programming tool. It enables the learners to perform data analysis on a real world scenario using appropriate tools.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Illustrate the mathematical concepts for data analytics (Cognitive Knowledge Level: Apply)
CO2	Explain the basic concepts of data analytics (Cognitive Knowledge Level: Understand)
CO3	Illustrate various predictive and descriptive analytics algorithms (Cognitive Knowledge Level: Apply)
CO4	Describe the key concepts and applications of Big Data Analytics (Cognitive Knowledge Level: Understand)
CO5	Demonstrate the usage of Map Reduce paradigm for Big Data Analytics (Cognitive Knowledge Level: Apply)
CO6	Use R programming tool to perform data analysis and visualization (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓	✓							✓
CO6	✓	✓	✓	✓	✓							✓

Abstract POs Defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	10 marks
Continuous Assessment Tests (Average of Series Tests 1& 2)	25 marks
Continuous Assessment Assignment	15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question

from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which students should answer any one. Each question can have a maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Mathematics for Data Analytics)

Descriptive statistics - Measures of central tendency and dispersion, Association of two variables - Discrete variables, Ordinal and Continuous variable, Probability calculus - probability distributions, Inductive statistics - Point estimation, Interval estimation, Hypothesis Testing - Basic definitions, t-test

Module - 2 (Introduction to Data Analytics)

Introduction to Data Analysis - Analytics, Analytics Process Model, Analytical Model Requirements. Data Analytics Life Cycle overview. Basics of data collection, sampling, preprocessing and dimensionality reduction

Module - 3 (Predictive and Descriptive Analytics)

Supervised Learning - Classification, Naive Bayes, KNN, Linear Regression. Unsupervised Learning - Clustering, Hierarchical algorithms – Agglomerative algorithm, Partitional algorithms - K- Means. Association Rule Mining - Apriori algorithm

Module - 4 (Big Data Analytics)

Big Data Overview – State of the practice in analytics, Example Applications - Credit Risk Modeling, Business Process Analytics. Big Data Analytics using Map Reduce and Apache Hadoop, Developing and Executing a HadoopMapReduce Program.

Module - 5 (R programming for Data Analysis)

Overview of modern data analytic tools. Data Analysis Using R - Introduction to R - R Graphical User Interfaces, Data Import and Export, Attribute and Data Types, Descriptive Statistics, Exploratory Data Analysis - Visualization Before Analysis, Dirty Data, Visualizing a Single Variable, Examining Multiple Variables, Data Exploration Versus Presentation, Statistical Methods for Evaluation

Text Book

1. Bart Baesens," Analytics in a Big Data World: The Essential Guide to Data Science and its Business Intelligence and Analytic Trends", John Wiley & Sons, 2013.
2. David Dietrich, "EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data", John Wiley & Sons, 2015.
3. Jaiwei Han, MichelineKamber, "Data Mining Concepts and Techniques", Elsevier, 2006.
4. Christian Heumann and Michael Schomaker, "Introduction to Statistics and DataAnalysis", Springer, 2016

References

1. Margaret H. Dunham, Data Mining: Introductory and Advanced Topics. Pearson, 2012.
2. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.

Course Level Assessment Questions**Course Outcome 1 (CO1):**

1. Explain the measures of central tendency.
2. Drive the mean and variance of normal distribution.
3. Collect sample data associated with a real world scenario, and identify central tendency and dispersion measures. Explain your inferences.

Course Outcome 2 (CO2):

1. Explain the life cycle of Data Analytics.
2. Discuss in detail the relevance of data sampling.

Course Outcome 3 (CO3):

1. The following table shows the midterm and final exam marks obtained for students in a database course.

X (Midterm exam)	Y (Final exam)
72	84
50	63

81	77
74	78
94	90
86	75
59	49
83	79
65	77
33	52
88	74
81	90

- a) Use the method of least squares to find an equation for the prediction of a student's final exam marks based on the student's midterm grade in the course.
 - b) Predict the final exam marks of a student who received an 86 on the midterm exam.
2. Perform knn classification on the following dataset and predict the class for the data point X (P1 = 3, P2 =7), assuming the value of k as 3.

P1	P2	Class
7	7	False
7	4	False
3	4	True
1	4	True

Course Outcome 4 (CO4):

1. List down the characteristics of Big Data.
2. Illustrate process discovery task in business analytics using the scenario of insurance claim handling process. Draw the annotated process map.

Course Outcome 5 (CO5):

1. Explain how fault tolerance is achieved in HDFS.
2. Write down the pseudocode for Map and Reduce functions to solve any one data analytic problem.

Course Outcome 6 (CO6):

1. Illustrate any three R functions used in data analytics.
2. Explain the different categories of attributes and data types in R.

Model Question Paper

QP CODE:

Reg No: _____

Name : _____

PAGES : 4

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 322

Course Name: Data Analytics

Max.Marks :100

Duration: 3 Hrs

PART A

(Answer all Questions. Each question carries 3 Marks)

1. Outline the errors that arise in hypothesis testing.
2. The number of members of a millionaires' club were as follows:

Year	2011	2012	2013	2014	2015	2016
Members	23	24	27	25	30	28

- (a)What is the average growth rate of the membership?
- (b)Based on the results of (a), how many members would one expect in 2018?

3. List and explain any two methods for dealing with missing values in a dataset.
4. Consider the following data (in increasing order) for the attribute age: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70. Sketch an example for stratified sampling using samples of size 5 and the strata “youth,” “middle-aged,” and “senior.”
5. Why is k nearest neighbor classifier called a lazy learner?
6. Find the absolute support, relative support and confidence of the rule (bread => jam) in the following set of transactions
 T1 {bread, butter}, T2 {bread, jam, milk}
 T3 {Milk, curd}, T4 {bread, jam}
7. Explain the 3 Vs of Big Data.
8. Discuss the application of big data analytics in credit risk modeling.
9. Why is Exploratory Data Analysis important in business application ?
10. Explain how box plots be used for data summarization.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Illustrate the Maximum Likelihood Estimation of Bernoulli distribution. (8)
- (b) A hiking enthusiast has a new app for his smartphone which summarizes his hikes by using a GPS device. Let us look at the distance hiked (in km) and maximum altitude (in m) for the last 10 hikes: (6)

Distance	12.5	29.9	14.8	18.7	7.6	16.2	16.5	27.4	12.1	17.5
Altitude	342	1245	502	555	398	670	796	912	238	466

Calculate the arithmetic mean and median for both distance and altitude.

OR

12. (a) Explain the steps in conducting a hypothesis test. (8)
- (b) A total of 150 customers of a petrol station are asked about their satisfaction with their car and motorbike insurance. The results are summarized below: Determine and interpret Pearson's χ^2 statistic and Cramer's V. (6)

	Satisfied	Unsatisfied	Total
Car	33	25	58
Car (Diesel engine)	29	31	60
Motor bike	12	20	32
Total	74	76	150

13. (a) Explain the data analytical process model. (8)
- (b) Discuss the methods for handling noisy data. Consider the following sorted data for price (in dollars) 4, 8, 15, 21, 21, 24, 25, 28, 34. (6)
 Illustrate smoothing by bin means and bin boundaries

OR

14. (a) a) What is the need for sampling in data analytics? Discuss the different sampling techniques. (8)
- (b) Use these methods to *normalize* the following group of data: (6)
 200, 300, 400, 600, 1000
 (i) min-max normalization by setting $min = 0$ and $max = 1$
 (ii) z-score normalization
 (iii) normalization by decimal scaling
15. (a) A database has five transactions. Let min_sup be 60% and min_conf be 80%. (6)

<i>TID</i>	<i>items_bought</i>
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I, E}

(a) Find all frequent itemsets using Apriori algorithm (10)

(b) Generate strong association rules from any one 3 itemset. (4)

OR

16. (a) Explain agglomerative hierarchical clustering with an example. (8)

(b) Suppose that the data mining task is to cluster points (with (x, y) representing location) into three clusters, where the points are A1(2,10), A2 (2,5), A3 (8,4), B1 (5,8), B2 (7,5), B3 (6,4), C1(1,2), C2 (4,9). The distance function is Euclidean distance. Suppose initially we assign A1, B1, and C1 as the center of each cluster, respectively. Use the k-means algorithm to show only

(a) The three cluster centers after the first round of execution.

(b) The final three clusters.

17. (a) Illustrate the working of a Map Reduce program with example. (8)

(b) Explain the data analytic architecture with a diagram. (6)

OR

18. (a) Discuss the architecture of HDFS and its features. (8)

(b) Illustrate the use of big data analytics in credit risk modeling. (6)

19. (a) List and explain the R functions used in descriptive statistics. (8)

(b) Explain hypothesis testing using ANOVA. (6)

OR

20. (a) Discuss the data visualization for multiple variables in R (8)

(b) Describe the R functions used for cleaning dirty data. (6)

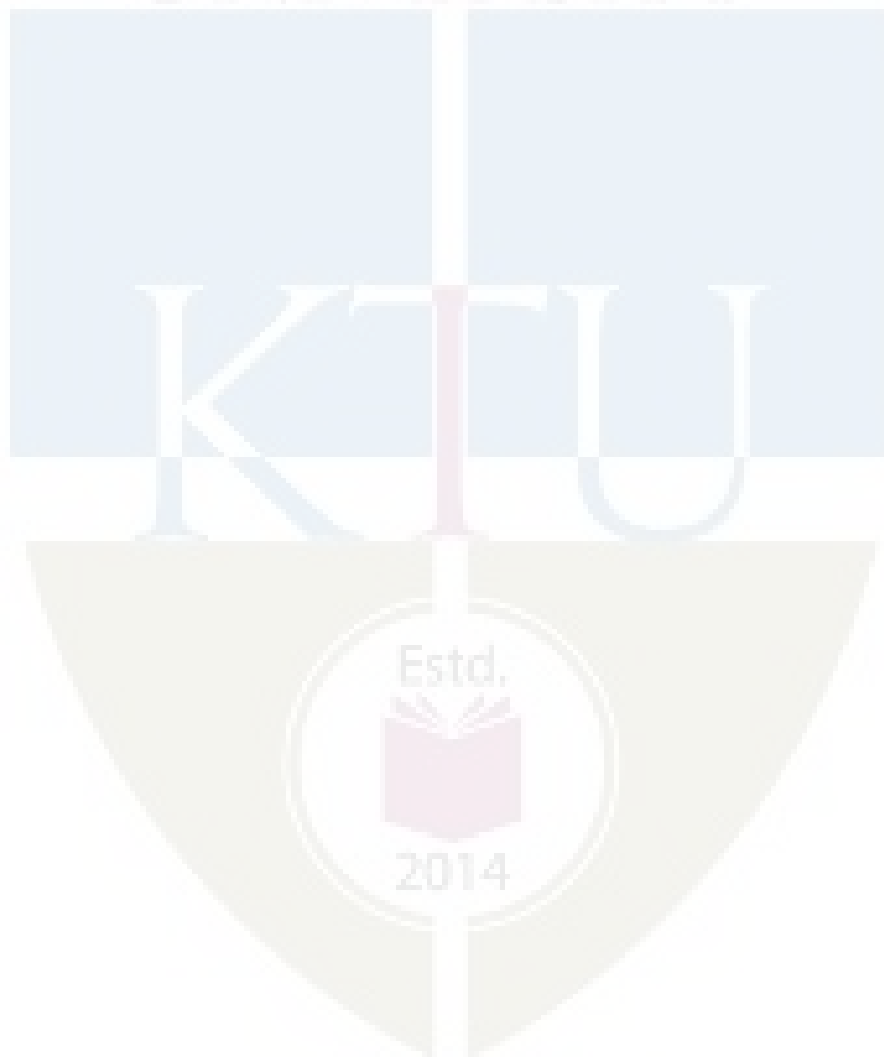
(5 x 14 = 70)

Teaching Plan

No	Contents	No of Lecture Hrs
Module – 1(Mathematics for Data Analytics) (7 hrs)		
1.1	Descriptive statistics - Measures of central tendency	1
1.2	Measures of dispersion	1
1.3	Association of two variables - Discrete Variables	1
1.4	Association of two variables - Ordinal and Continuous variable	1
1.5	Probability calculus - Probability distributions	1
1.6	Inductive statistics - Point estimation, Interval estimation	1
1.7	Hypothesis Testing - Basic definitions, t-test	1
Module – 2 (Introduction to Data Analytics) (6 hrs)		
2.1	Introduction to Data Analysis –Analytics, Analytics process model	1
2.3	Analytical model requirements	1
2.4	Data Analytics Life Cycle overview	1
2.5	Basics of data collection	1
2.6	Basics of sampling and preprocessing	1
2.7	Dimensionality reduction	1
Module - 3 (Predictive and Descriptive Analytics) (8 hrs)		
3.1	Supervised Learning, Naive Bayes classification	1
3.2	KNN algorithm	1

3.3	Linear Regression	1
3.4	Unsupervised Learning- Clustering	1
3.5	Hierarchical algorithms Agglomerative algorithm	1
3.6	Partitional algorithms -K- Means	1
3.7	Association Rule Mining	1
3.8	Apriori algorithm	1
Module - 4 (Big Data Analytics) (7 hrs)		
4.1	Big Data Overview – State of the practice in analytics.	1
4.2	Example Applications - Credit Risk Modeling	1
4.3	Business Process Analytics.	1
4.4	Big Data Analytics using Map Reduce and Apache Hadoop	1
4.5	Big Data Analytics using Map Reduce and Apache Hadoop	1
4.6	Developing and Executing a Hadoop MapReduce Program	1
4.7	Developing and Executing a Hadoop MapReduce Program	1
Module - 5 (R programming for Data Analysis) (8 hrs)		
5.1	Overview of modern data analytic tools, Introduction to R, R Graphical User Interfaces	1
5.2	Data Import and Export, Attribute and Data Types	1

5.3	Descriptive Statistics	1
5.4	Exploratory Data Analysis, Visualization Before Analysis	1
5.5	Dirty Data, Visualizing a Single Variable	1
5.6	Examining Multiple Variable	1
5.7	Data Exploration Versus Presentation	1
5.8	Statistical Methods for Evaluation	1



CST 332	FOUNDATIONS OF SECURITY IN COMPUTING	Category	L	T	P	Credit	Year Of Introduction
		PEC	2	1	0	3	2019

Preamble: The purpose of this course is to create awareness among learners about the fundamentals of security and number theory. This course covers Integer & Modular Arithmetic, Primes & Congruences, Discrete Logarithms & Elliptic Curve Arithmetic and an overview of computer security. The concepts covered in this course enable the learners in effective use of cryptographic algorithms and to identify the security threats in computing.

Prerequisite: A sound knowledge in Mathematics, Discrete Computational Structures, Operating Systems and Database Systems.

Course Outcomes: After the completion of the course, the student will be able to

CO1	Illustrate the operations and properties of algebraic structures, integer arithmetic and modular arithmetic. (Cognitive Knowledge Level: Understand)
CO2	Use the concepts of prime numbers and factorization for ensuring security in computing systems (Cognitive Knowledge Level: Apply)
CO3	Illustrate the concepts of Linear Congruence, Primitive Roots, Discrete Logarithms and Elliptic Curve Arithmetic (Cognitive Knowledge Level: Apply)
CO4	Summarize the threats and attacks related to computer and program security (Cognitive Knowledge Level: Understand)
CO5	Outline the key aspects of operating system and database security (Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	☑	☑	☑									☑
CO2	☑	☑	☑	☑								☑
CO3	☑	☑	☑	☑								☑
CO4	☑	☑	☑			☑		☑				☑
CO5	☑	☑	☑			☑		☑				☑

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (%)	Test 2 (%)	End Semester Examination (%)
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			

Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Test : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus**Module-1 (Modular Arithmetic)**

Integer arithmetic - Integer division, Divisibility, Greatest Common Divisor (GCD), Euclid's algorithm for GCD, Extended Euclid's algorithm, Linear Diophantine Equations. Modular

arithmetic - Operations, Properties. Algebraic structures - Groups, Rings, Fields, Finite fields, $GF(p)$, $GF(2^n)$.

Module-2 (Prime Numbers and Factorization)

Prime numbers - Prime numbers and prime-power factorization, Fermat and Mersenne primes, Fermat's theorem, Applications, Euler's theorem, Euler's totient function, Applications. Primality testing – Deterministic algorithms and Probabilistic algorithms. Factorization - Fermat's factorization, Pollard p-1 method.

Module-3 (Linear Congruence, Primitive Roots and Elliptic Curve Arithmetic)

Linear congruence - Simultaneous linear congruence, Chinese Remainder Theorem (CRT). Congruence with a prime - Power modulus, Arithmetic modulo p, Pseudoprimes and Carmichael numbers, Solving congruence modulo prime powers. Primitive roots - Existence of primitive roots for primes, Discrete logarithms. Elliptic curve arithmetic – Prime curves, Binary curves, Addition of two points, Multiplication of a point by a constant.

Module-4 (Computer and Program Security)

Introduction to computer security – Threats, Vulnerabilities, Controls. Browser attack types, Web attacks targeting users, Email attack types. Introduction to program security - Non-malicious programming oversights, Malware.

Module-5 (Operating System and Database Security)

Operating system security – Security in operating system, Security in design of operating system. Database security – Security requirements of databases, Reliability and integrity, Database disclosure.

Text Books

1. Behrouz A Forouzan, Cryptography and Network Security, 3/e, Tata McGraw-Hill.
2. Charles P Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, Security in Computing, 5/e, Prentice Hall.
3. G.A. Jones & J.M. Jones, Elementary Number Theory, Springer UTM, 2007

References

1. William Stallings, Cryptography and Network Security Principles and Practices, 4/e, Pearson Ed.

Sample Course Level Assessment Questions**Course Outcome 1 (CO1):**

1. Find the n- bit word that is represented by the polynomial $x^2 + 1$ in $GF(2^5)$.
2. Solve the linear Diophantine equation $21x + 14y = 35$.

Course Outcome 2 (CO2):

1. Prove that a Carmichael number cannot be the product of two distinct primes.
2. Use the Pollard p-1 method to find a factor of 57247159 with the bound $B=8$.

Course Outcome 3 (CO3):

1. Find an integer that has a remainder of 3 when divided by 7 and 13, but is divisible by 12.
2. In the elliptic curve $E(1,2)$ over the field $GF(11)$, find the equation of the curve and all the points on the curve.

Course Outcome 4 (CO4):

1. List three controls that could be applied to detect or prevent off-by-one errors.
2. How does fake email messages act as spam?

Course Outcome 5 (CO5):

1. Discuss the importance of auditability and access control in database security.
2. Explain the various factors which can make data sensitive.

Model Question Paper

QP CODE:

PAGES: ____

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 332

Course Name : FOUNDATIONS OF SECURITY IN COMPUTING

Max Marks: 100

Duration: 3 Hours

PART A

(Answer All Questions. Each question carries 3 marks)

1. List the four properties of divisibility with examples.
2. Find gcd (401,700) using Euclid's algorithm.
3. Use Fermat's Little theorem to show that 91 is not a prime.
4. If m is relatively prime to n , show that $\Phi(mn) = \Phi(m) \Phi(n)$.
5. Solve the congruence relation $103x \equiv 57 \pmod{211}$.
6. Find a solution for the congruence $3x \equiv 5 \pmod{7^3}$
7. What are the problems created by an off-by-one error?
8. How does a clickjacking attack succeed?
9. Explain the significance of correctness and completeness in the design of operating systems.
10. How does the two-phase update technique help the database manager in handling failures? **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) For the group $G = \langle \mathbb{Z}_6^*, x \rangle$, prove that it is an Abelian group. Also show the result of 5×1 and $1 \div 5$. (6)

(b) Find a particular and the general solution to the following linear Diophantine equations. (8)

i) $19x + 13y = 20$ ii) $40x + 16y = 88$

OR

12. (a) Describe the properties of modular arithmetic and modulo operator. (6)

(b) Using Extended Euclidean algorithm, find the multiplicative inverse of (i) 131 in \mathbb{Z}_{180} and (ii) 23 in \mathbb{Z}_{100} . (8)

13. (a) State and prove Fermat's theorem. (6)

(b) Explain Fermat's factorization method and use it to factor 809009. (8)

OR

14. (a) Define Euler's totient function. Prove that, $\phi(pq) = (p-1)(q-1)$ where p and q are prime numbers. (7)

(b) Define Fermat primes. Show that any two distinct Fermat numbers are relatively prime. (7)

15. (a) Using Chinese Remainder Theorem, solve the system of congruence, $x \equiv 2 \pmod{3}$, $x \equiv 3 \pmod{5}$, $x \equiv 2 \pmod{7}$. (7)

(b) Define Carmichael number and show that a Carmichael number must be the product of at least three distinct primes. (7)

OR

16. (a) For the group $G = \langle \mathbb{Z}_{19}^*, x \rangle$, find the primitive roots in the group. (6)

(b) Consider the elliptic curve $y^2 = x^3 + x + 1$ defined over \mathbb{Z}_{23} . If $P = (3, 10)$ and $Q = (9, 7)$ are two points on the elliptic curve, find $2P$ and $P + Q$. (8)

17. (a) Distinguish the terms vulnerability, threat and control. (4)

(b) With the help of suitable examples, explain the security problems created by incomplete mediation and time-of-check to time-of-use. (10)

OR

18. (a) Differentiate between man-in-the-browser attack and page-in-the-middle attack. (4)

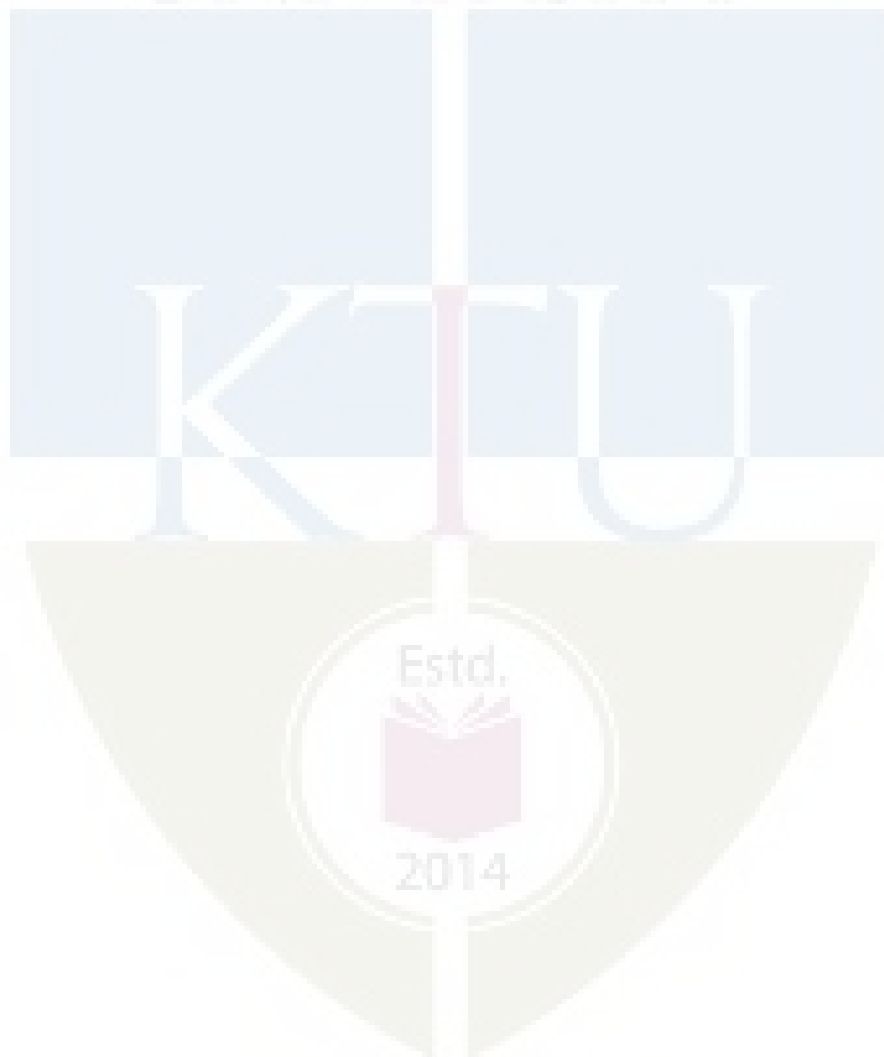
- (b) Explain the four aspects of malicious code infection. (10)
19. (a) List any six computer security related functions addressed by operating systems. (6)
- (b) How does a kernelized design support in enforcing security mechanisms? (8)
- OR**
20. (a) Explain any four security requirements of databases. (4)
- (b) How can database disclosure be prevented? With the help of suitable examples, explain any six types of disclosure. (10)

Teaching Plan

No	Contents	No.of Lecture Hrs
Module-1 (Modular Arithmetic) (6 hrs)		
1.1	Integer arithmetic, Integer division, Divisibility, Greatest Common Divisor (GCD)	1
1.2	Euclid's algorithm for GCD, Extended Euclid's algorithm	1
1.3	Linear Diophantine Equations	1
1.4	Modular arithmetic operations, Properties of modular arithmetic	1
1.5	Groups, Rings and Fields	1
1.6	Finite fields – $GF(p)$, $GF(2^n)$	1
Module-2 (Prime Numbers and Factorization) (7 hrs)		
2.1	Prime numbers and prime-power factorization	1
2.2	Fermat and Mersenne primes	1
2.3	Fermat's theorem, Applications – Exponentiation, Multiplicative inverse	1
2.4	Euler's theorem, Euler's totient function, Applications	1
2.5	Primality testing – Deterministic algorithms – Divisibility algorithm	1

2.6	Primality testing – Probabilistic algorithms-Fermat test, Square root test, Miller - Rabin test	1
2.7	Factorization - Fermat’s factorization, Pollard p-1 method	1
Module-3 (Linear Congruence, Primitive Roots and Elliptic Curve Arithmetic) (7 hrs)		
3.1	Linear congruence, Simultaneous linear congruence	1
3.2	Chinese Remainder Theorem (CRT)	1
3.3	Congruence with a Prime-Power Modulus, Arithmetic modulo p	1
3.4	Pseudo-primes and Carmichael numbers	1
3.5	Solving congruence modulo prime powers	1
3.6	Primitive roots, Existence of primitive roots for primes, Discrete logarithms	1
3.7	Elliptic curve arithmetic – Prime curves, Binary curves, Addition of two points, Multiplication of a point by a constant	1
Module-4 (Computer and Program Security) (7 hrs) (Text book2: Chapters 1, 3, 4)		
4.1	Threats, Vulnerabilities, Controls	1
4.2	Browser attack types	1
4.3	Web attacks targeting users	1
4.4	Email attack types	1
4.5	Non-malicious programming oversights (Lecture 1)	1
4.6	Non-malicious programming oversights (Lecture 2)	1
4.7	Malware – Four aspects of infection	1
Module-5 (Operating System and Database Security) (8 hrs)(Text book2: Chapters 5, 7)		
5.1	Security in operating system (Lecture 1)	1
5.2	Security in operating system (Lecture 2)	1
5.3	Security in design of operating system (Lecture 1)	1

5.4	Security in design of operating system (Lecture 2)	1
5.5	Security requirements of databases	1
5.6	Reliability & integrity	1
5.7	Database disclosure (Lecture 1)	1
5.8	Database disclosure (Lecture 2)	1



CST 342	AUTOMATED VERIFICATION	Category	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0		3

Preamble: This course is intended to impart the basic theory and algorithm for an automatic verification process namely model checking. This course covers finite-state modelling of hardware/software, linear-time properties, classification of linear-time properties, Linear Temporal Logic (LTL) - a formal language for property specification, LTL model checking algorithm and model checking case studies. This course enables the learners to prove correctness of a hardware/software used in safety critical systems in domains such as avionics, health care and automotive.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO1	Illustrate an application for model checking. (Cognitive Knowledge Level: Understand)
CO2	Describe finite-state modelling for hardware and software. (Cognitive Knowledge Level: Understand)
CO3	Identify linear-time properties required to represent the requirements of a system. (Cognitive Knowledge Level: Apply)
CO4	Specify a given linear-time property in Linear Temporal Logic (LTL). (Cognitive Knowledge Level: Apply)
CO5	Perform LTL model checking using the tool Symbolic Analysis Laboratory (SAL). (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>
CO2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>
CO3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>
CO4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>
CO5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Marks)	Test 2 (Marks)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Test : 25 marks

Continuous Assessment Assignment : 15 marks (Out 15, 10 marks shall be given for a model checking project to be implemented in SAL.)

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each

question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction to Model Checking)

System Verification – Hardware and Software Verification, Model Checking, Characteristics of Model Checking.

Transition Systems – Transition System, Direct Predecessors and Successors, Terminal State, Deterministic Transition System.

Executions - Execution Fragment, Maximal and Initial Execution Fragment, Execution, Reachable States.

Module - 2 (Linear Time Properties)

Linear-Time (LT) Properties - Deadlock. Linear-Time Behavior - Paths and State Graph, Path Fragment, Maximal and Initial Path Fragment, Path. Traces - Trace and Trace Fragment, LT Properties - LT Property, Satisfaction Relation for LT Properties, Trace Equivalence and LT Properties. Safety Properties and Invariants - Invariants, Safety Properties, Trace Equivalence and Safety properties. Liveness Properties - Liveness Property, Safety vs. Liveness Properties. Fairness - Fairness, Unconditional, Weak and Strong Fairness, Fairness Strategies, Fairness and Safety. (Definition and examples only for all topics - no proof required).

Module - 3 (Regular Properties)

Regular Properties - Model Checking Regular Safety properties - Regular Safety property, Verifying Regular Safety Properties. Automata on Infinite Words - ω -Regular Languages and Properties, Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA),

Generalised Buchi Automata (Definitions only). Model Checking ω -Regular Properties - Persistence Properties and Product, Nested Depth-First Search (Only algorithms required).

Module - 4 (Linear Time Logic)

Linear Temporal Logic (LTL) - Syntax, Semantics, Equivalence of LTL Formulae, Weak Until, Release and Positive Normal Form, Fairness, Safety and Liveness in LTL (Definitions only). Automata Based LTL Model Checking (Algorithms and examples only).

Module - 5 (Model Checking in SAL)

Introduction - Introduction to the tool Symbolic Analysis Laboratory (SAL).

The Language of SAL - The expression language, The transition Language, The module language, SAL Contexts.

SAL Examples - Mutual Exclusion, Peterson's Protocol, Synchronous Bus Arbiter, Bounded Bakery protocol, Bakery Protocol, Traffic Signalling System.

Text Books

1. Christel Baier and Joost-Pieter Katoen, Principles of Model Checking, The MIT Press. (Modules 1 - 4)
2. Leonardo de Moura, Sam Owre and N. Shankar, The SAL Language Manual, SRI International (<http://sal.csl.sri.com/doc/language-report.pdf>, Chapters 1, 3, 4, 5, 6, 7) (Module 5)

Reference Materials

1. SAL Examples (<http://sal.csl.sri.com/examples.shtml>) (Module 5)

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Illustrate how model checking can make a system design reliable, based on a required set of properties/constraints.

Course Outcome 2 (CO2):

1. Consider a message delivery system. The sender s is trying to send a series of messages to the receiver r in such a way that the $(i+1)^{st}$ message is sent only after the i^{th} message is delivered. There is a possibility of error in sending a message and in that case, s keeps on

trying until it is able to send the message. Show a finite state transition system modeling this system.

Course Outcome 3 (CO3):

1. Consider a shared memory segment s protected using a mutex lock variable m . Two processes p_1 and p_2 are trying to access s . List the Linear Time properties of the system which will ensure safety, liveness and fairness.

Course Outcome 4 (CO4):

1. Show the LTL specifications of the safety, liveness and fairness properties listed for the assessment question given in CO3.

Course Outcome 5 (CO5):

1. Model the system mentioned in the question given in CO3 in SAL and verify that the system is correct with respect to the LTL properties shown as the answer for CO4.

Model Question paper**QP CODE:****PAGES: 3**

Reg No: _____

Name : _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR****Course Code: CST342****Course Name: Automated Verification****Max.Marks:100****Duration: 3 Hours****PART A****Answer all questions. Each question carries 3 marks.**

1. Define model checking. Show the schematic diagram of the model checking approach.
2. Show a transition system modeling a coffee/Tea vending machine.

3. Define invariant as a Linear Time (LT) property. Give an example
4. List any three Linear Time properties in the Mutual Exclusion problem of processes.
5. Illustrate the construction of a product automaton from two automata.
6. Differentiate between Deterministic Buchi Automaton and Non-deterministic Buchi Automaton.
7. Specify the following statements about traffic lights in Linear Temporal Logic (LTL).
 - a. Once red, the light can not become green immediately.
 - b. Once red, the light always becomes green eventually after being yellow for some time.
8. What is Positive Normal Form (PNF) in LTL? Give an example.
9. List any three applications of the tool Symbolic Analysis Laboratory (SAL).
10. What is a SAL context? Give an example.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain in detail the various phases of the model checking process. (8)
 - (b) Explain the strengths and weaknesses of model checking. (6)
- OR**
2014
12. (a) Define and illustrate the following terms of a transition system. (14)
 - a. Execution Fragment
 - b. Maximal and Initial Execution Fragment
 - c. Execution
 - d. Reachable States

13. (a) With an example, explain the satisfaction relation for LT properties. (7)
- (b) What is trace equivalence in Transition Systems? Give an example to show that if two transition systems satisfy the trace equivalence property, then they satisfy the same set of LT properties. (7)
- OR**
14. (a) Give the transition system for the fault tolerant variant of the dining philosophers problem. (4)
- (b) With a suitable example, explain the algorithms to check whether a Transition System satisfies an invariant or not. (10)
15. (a) Explain Regular Safety Properties with a suitable example. (7)
- (b) Illustrate an algorithm for verifying Regular Safety Properties. (7)
- OR**
16. (a) Explain ω -Regular Properties. (4)
- (b) Illustrate how ω -Regular Properties are verified. (10)
17. (a) Explain the syntax of Linear Temporal Logic (LTL). (7)
- (b) Explain the semantics of LTL. (7)
- OR**
18. (a) With an example, give the difference between until and weak until in LTL. (4)
- (b) With a suitable example, explain automata based LTL model checking. (10)
19. (a) Explain Peterson's protocol. What are the LTL properties to be verified to ensure its correctness? (8)
- (b) Write a SAL script for the verification of Peterson's protocol. (6)

OR

20. (a) Show the SAL model corresponding to Bakery protocol. (8)

(b) List any three Linear Time properties of this model and show their LTL (6)

Teaching Plan

Module 1 (Introduction to Model Checking)		4 Hours
1.1	System Verification – Hardware and Software Verification, Model Checking, Model Checking	1 Hour
1.2	Transition Systems – Transition System, Direct Predecessors and Successors, Terminal State, Deterministic Transition System	1 Hour
1.3	Executions - Execution Fragment, Maximal and Initial Execution Fragment	1 Hour
1.4	Execution, Reachable States	1 Hour
Module 2 (Linear Time Properties)		8 Hours
2.1	Linear-Time (LT) Properties - Deadlock	1 Hour
2.2	Linear-Time Behavior - Paths and State Graph, Path Fragment, Maximal and Initial Path Fragment, Path	1 Hour
2.3	Traces - Trace and Trace Fragment	1 Hour
2.4	LT Property, Satisfaction Relation for LT Properties, Trace Equivalence and LT Properties	1 Hour
2.5	Invariants	1 Hour
2.6	Safety Properties, Trace Equivalence and Safety properties	1 Hour
2.7	Liveness Property, Safety vs. Liveness Properties	1 Hour
2.8	Fairness, Unconditional, Weak and Strong Fairness, Fairness Strategies, Fairness and Safety	1 Hour
Module 3 (Regular Properties)		9 Hours
3.1	Regular Properties - Model Checking Regular Safety properties - Regular Safety property	1 Hour
3.2	Verifying Regular Safety Properties	1 Hour
3.3	Automata on Infinite Words - ω -Regular Languages and Properties	2 Hour

3.4	Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA), Generalised Buchi Automata	1 Hour
3.5	Model Checking ω -Regular Properties - Persistence Properties and Product - Lecture 1	1 Hour
3.6	Persistence Properties and Product - Lecture 2	1 Hour
3.7	Nested Depth-First Search (Lecture 1)	1 Hour
3.8	Nested Depth-First Search (Lecture 2)	1 Hour
Module 4 (Linear Time Logic)		7 Hours
4.1	Linear Temporal Logic – Linear Temporal Logic (LTL) - Syntax	1 Hour
4.2	Semantics - Lecture 1	1 Hour
4.3	Equivalence of LTL Formulae, Weak Until	1 Hour
4.4	Release and Positive Normal Form	1 Hour
4.5	Fairness, Safety and Liveness in LTL	1 Hour
4.6	Automata Based LTL Model Checking (Lecture 1)	1 Hour
4.7	Automata Based LTL Model Checking (Lecture 2)	1 Hour
Module 5 (Model Checking in SAL)		7 Hours
5.1	Introduction - Introduction to the tool Symbolic Analysis Laboratory (SAL).	1 Hour
5.2	The Language of SAL - The expression language, The transition Language	1 Hour
5.3	The module language, SAL Contexts.	1 Hour
5.4	SAL Examples - Mutual Exclusion	1 Hour
5.5	Peterson's Protocol, Synchronous Bus Arbiter	1 Hour
5.6	Bounded Bakery protocol, Bakery Protocol	1 Hour
5.7	Traffic Signalling System	1 Hour

CST 362	PROGRAMMING IN PYTHON	Category	L	T	P	Credit	Year of Introduction
		PEC	2	1	0	3	2019

Preamble: The objective of the course is to equip the learners to develop multi-module software solutions for real world computational problems using Python. It encompasses the Python programming environment, syntax, data representations, intermediate level features, GUI programming, Object Oriented Programming and data processing. This course lays the foundation to develop modular software solutions including complex interactive applications, network applications, and data-driven intelligent applications.

Prerequisite: Basic knowledge in Computational Problem Solving, A course in any programming language.

Course Outcomes: After the completion of the course the student will be able to

CO1	Write, test and debug Python programs (Cognitive Knowledge level: Apply)
CO2	Illustrate uses of conditional (if, if-else and if-elif-else) and iterative (while and for) statements in Python programs. (Cognitive Knowledge level: Apply)
CO3	Develop programs by utilizing the Python programming constructs such as Lists, Tuples, Sets and Dictionaries. (Cognitive Knowledge level: Apply)
CO4	Develop graphical user interface for solutions using Python libraries. (Cognitive Knowledge level: Apply)
CO5	Implement Object Oriented programs with exception handling. (Cognitive Knowledge level: Apply)
CO6	Write programs in Python to process data stored in files by utilizing Numpy, Matplotlib, and Pandas. (Cognitive Knowledge level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓		✓							✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓							✓
CO5	✓	✓	✓	✓	✓							✓
CO6	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation

#PO	Broad PO	#PO	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (Marks in percentage)	Test 2 (Marks in percentage)	End Semester Examination Marks
Remember	20	20	20

Understand	35	35	35
Apply	45	45	45
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Test : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B

contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 sub-divisions and carries 14 marks.

SYLLABUS

Module -1 (Programming Environment and Python Basics) (6 hours)

Getting started with Python programming – Interactive shell, IDLE, iPython Notebooks, Detecting and correcting syntax errors, How Python works. The software development process – A case study. Basic coding skills – strings, assignment, and comments, Numeric data types and character sets, Expressions, Using inbuilt functions and modules. Control statements – Iteration with for/while loop, Formatting text for output, A case study, Selection structure (if-else, switch-case), Conditional iteration with while, A case study, Testing control statements, Lazy evaluation.

Module -2 (Building Python Programs) (8 hours)

Strings and text files – Accessing characters, substrings, Data encryption, Strings and number system, String methods, Text files, A case study on text analysis. Design with Functions – Functions as Abstraction Mechanisms, Problem solving with top-down design, Design with recursive functions, Managing a program’s namespace, Higher-Order Functions. Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension. Work with tuples. Sets. Work with dates and times, A case study with lists. Dictionaries - Dictionary functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries, reverse lookup. Case Study – Data Structure Selection.

Module -3 (Graphics) (7 hours)

Graphics – Terminal-based programs, Simple Graphics using Turtle, Operations, 2D Shapes, Colors and RGB Systems, A case study. Image Processing – Basic image processing with inbuilt functions. Graphical User Interfaces – Event-driven programming, Coding simple GUI-based programs : Windows, Labels, Displaying images, Input text entry, Popup dialog boxes, Command buttons, A case study.

Module -4 (Object Oriented Programming) (7 hours)

Design with classes - Objects and Classes, Methods, Instance variables, Constructor, Accessor and Mutator, Data-Modeling Examples, Structuring classes with inheritance and polymorphism. Abstract classes, Interfaces, Exceptions - Handle a single exception, handle multiple exceptions.

Module -5 (Data Processing) (9 hours)

The os and sys modules, NumPy - Basics, Creating arrays, Arithmetic, Slicing, Matrix Operations, Random numbers. Plotting and visualization. Matplotlib - Basic plot, Ticks, Labels, and Legends. Working with CSV files. – Pandas - Reading, Manipulating, and Processing Data. Introduction to Micro services using Flask.

Text Books:

1. Kenneth A Lambert., Fundamentals of Python : First Programs, 2/e, Cengage Publishing, 2016
2. Wes McKinney, Python for Data Analysis, 2/e, Shroff / O'Reilly Publishers, 2017
3. Flask: Building Python web services, Jack Stouffer, Shalabh Aggarwal, Gareth Dwyer, PACKT Publishing Limited, 2018

Reference Books:

1. Zed A Shaw, Learn Python 3 The Hard Way, Addison-Wesley, 2017
2. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2/e, Schroff, 2016
3. Michael Urban and Joel Murach, Python Programming, Shroff/Murach, 2016
4. Charles Severance. Python for Informatics: Exploring Information,

Sample Course Level Assessment Questions**Course Outcome1(CO1):**

1. What is type conversion? How is it done in Python?
2. Write a note on the Python editors.

Course Outcome 2(CO2):

1. Write a Python program which takes a positive integer **n** as input and finds the sum of cubes all positive even numbers less than or equal to the number.
2. What is printed when the below code is executed?

```
mysum = 0
for i in range(5, 11, 2):
    mysum += i
    if mysum == 5:
        break
    mysum += 1
print(mysum)
```

What would be the output if 'break' is replaced with 'continue' in the above code fragment?

Course Outcome 3(CO3):

1. Given is a list of words, *wordlist*, and a string, *name*. Write a Python function which takes *wordlist* and *name* as input and returns a tuple. The first element of the output tuple is the number of words in the *wordlist* which have *name* as a substring in it. The second element of

the tuple is a list showing the index at which the *name* occurs in each of the words of the *wordlist* and a 0 if it doesn't occur.

2. What is the value of L after you run the code below?

```
L = ["life", "answer", 42, 0]
for thing in L:
    if thing == 0:
        L[thing] = "universe"
    elif thing == 42:
        L[1] = "everything"
```

Course Outcome 4(CO4):

1. A bouncy program is defined as follows – The program computes and displays the total distance traveled by a ball, given three inputs—the initial height from which it is dropped, its bounciness index, and the number of bounces. Given the inputs write a GUI-based program to compute the total distance traveled.
2. Write a Python program to find the quadrant of a point, say (x,y).

Course Outcome 5(CO5):

1. Write a Python program to implement the addition, subtraction, and multiplication of complex numbers using classes. Use constructors to create objects. The input to the program consist of real and imaginary parts of the complex numbers.
2. Explain inheritance in Python using suitable examples.

Course Outcome 6(CO6):

1. Given a file “auto.csv” of automobile data with the fields *index*, *company*, *body-style*, *wheel-base*, *length*, *engine-type*, *num-of-cylinders*, *horsepower*, *average-mileage*, and *price*, write python code to
 1. Clean and Update the CSV file
 2. Print total cars of all companies
 3. Find the average mileage of all companies
 4. Find the highest priced car of all companies.
2. Given two matrices A and B, write a program to find the product of A and B^T .

Model Question Paper

QP CODE:

PAGES:

Reg No: _____

Name: _____

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY SIXTH SEMESTER B.TECH
DEGREE EXAMINATION, MONTH & YEAR****Course Code: CST 362****Course name : PROGRAMMING IN PYTHON****Max Marks: 100****Duration: 3 Hours****PART-A****(Answer All Questions. Each question carries 3 marks)**

1. Write a Python program to reverse a number and also find the sum of digits of the number. Prompt the user for input.
2. Explain the concept of scope and lifetime of variables in Python programming language, with a suitable example.
3. Illustrate format specifiers and escape sequences with examples.
4. Compare tuples, lists, and dictionaries with examples.
5. Describe the following dictionary methods with an example.
i. *get()* ii. *Keys()* iii. *pop()* iv. *update()* v. *values()* vi. *items()*
6. Differentiate the terminal-based and GUI-based programming in Python.
7. What is polymorphism? Give an example in the context of OOP in Python.
8. How is exception handling accomplished in Python programs?
9. Explain the **os** and **os.path** modules in Python with examples. Also, discuss the *walk()* and *getcwd()* methods of the **os** module.
10. What are the important characteristics of CSV file format.

PART-B**(Answer any one full question from each module)**

11. (a) Write a Python code to check whether a given year is a leap year or not (6)
[An year is a leap year if it's divisible by 4 but not divisible by 100 except for those divisible by 400].

- (b) What are the possible errors in a Python program. Write a Python (8)
program to print the value of $2^{2n}+n+5$ for n provided by the user.

OR

12. (a) Write a Python program to find the value for $\sin(x)$ up to n terms using the (6)
series

$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad \text{where } x \text{ is in degrees}$$

- (b) Write a Python code to determine whether the given string is a Palindrome (8)
or not using slicing. Do not use any string function.

13. (a) Write a Python code to create a function called *list_of_frequency* that takes a (5)
string and prints the letters in non-increasing order of the frequency of their
occurrences. Use dictionaries.

- (b) Write a Python program to read a list of numbers and sort the list in a non- (9)
decreasing order without using any built in functions. Separate function
should be written to sort the list wherein the name of the list is passed as the
parameter.

OR

14. (a) Illustrate the following Set methods with an example. (6)
i. *intersection()* ii. *Union()* iii. *Issubset()* iv. *Difference()* v. *update()* vi.
discard()

- (b) Write a Python program to check the validity of a password given by the (8)
user.

The Password should satisfy the following criteria:

1. Contains at least one letter between **a** and **z**
2. Contains at least one number between **0** and **9**
3. Contains at least one letter between **A** and **Z**
4. Contains at least one special character from **!, #, @**
5. Minimum length of password: **6**

15. (a) Write a program to draw a hexagon using turtle. (5)
 (b) Write a note on the image processing function in Python. (9)

OR

16. (a) Describe the features of event driven programming. (4)
 (b) Write a GUI-based program that allows the user to convert temperature values between degrees Fahrenheit and degrees Celsius. The interface should have labeled entry fields for these two values. These components should be arranged in a grid where the labels occupy the first row and the corresponding fields occupy the second row. At start-up, the Fahrenheit field should contain 32.0, and the Celsius field should contain 0.0. The third row in the window contains two command buttons, labeled >>>> and <<<<. When the user presses the first button, the program should use the data in the Fahrenheit field to compute the Celsius value, which should then be output to the Celsius field. The second button should perform the inverse function. (10)
17. (a) How can a class be instantiated in Python? Write a Python program to express the instances as return values to define a class RECTANGLE with parameters *height*, *width*, *corner_x*, and *corner_y* and member functions to find center, area, and perimeter of an instance. (10)
 (b) Explain inheritance in Python. Give examples for each type of inheritance. (4)

OR

18. (a) Write a Python class named **Circle** constructed by a radius and two methods which will compute the area and the perimeter of a given circle (6)
 (b) Write Python program to create a class called as **Complex** and implement `__add__()` method to add two complex numbers. Display the result by overloading the + Operator. (8)
19. (a) Write a Python program to add two matrices and also find the transpose of the resultant matrix. (8)
 (b) Given a file “auto.csv” of automobile data with the fields *index*, *company*, *body-style*, *wheel-base*, *length*, *engine-type*, *num-of-cylinders*, *horsepower*, *average-mileage*, and *price*, write Python codes using Pandas to
 1) Clean and Update the CSV file
 2) Print total cars of all companies

- 3) Find the average mileage of all companies
- 4) Find the highest priced car of all companies.

OR

20. (a) Write Python program to write the data given below to a CSV file. (5)

SN	Name	Country	Contribution	Year
1	Linus Torvalds	Finland	Linux Kernel	1991
2	Tim Berners-Lee	England	World Wide Web	1990
3	Guido van Rossum	Netherlands	Python	1991

- (b) Given the sales information of a company as CSV file with the following fields *month_number, facecream, facewash, toothpaste, bathingsoap, shampoo, moisturizer, total_units, total_profit*. Write Python codes to visualize the data as follows (9)

- 1) Toothpaste sales data of each month and show it using a scatter plot
- 2) Face cream and face wash product sales data and show it using the bar chart

Calculate total sale data for last year for each product and show it using a Pie chart.

(14X5=70)

Teaching Plan

Module 1: Programming Environment and Python Basics		(6 hours)
1.1	Getting started with Python programming – Interactive shell, IDLE, iPython Notebooks, Detecting and correcting syntax errors, How Python works.	1 hour
1.2	The software development process – A case study.	1 hour
1.3	Basic coding skills – strings, assignment, and comments, Numeric data types and character sets	1 hour
1.4	Expressions, Using inbuilt functions and modules.	1 hour
1.5	Control statements – Definite Iteration with for loop, Formatting text for output, Selection structure (if-else, switch-case), Conditional iteration with	1 hour

	while loop, A case study	
1.6	Testing the control statements, Lazy evaluation.	1 hour
Module 2: Building Python Programs		(8 hours)
2.1	Strings – Accessing characters, substrings, Data encryption, Strings and number system, String methods,	1 hour
2.2	Text files, A case study on text analysis.	1 hour
2.3	Design with Functions – Functions as Abstraction Mechanisms, Problem solving with top-down design,	1 hour
2.4	Design with recursive functions, Managing a program’s namespace, Higher-Order Functions.	1 hour
2.5	Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension.	1 hour
2.6	Work with tuples. Sets. Work with dates and times, A case study with lists.	1 hour
2.7	Dictionaries - Dictionary functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries, reverse lookup.	1 hour
2.8	Case Study - Data Structure Selection.	1 hour
Module 3: Graphics		(7 hours)
3.1	Graphics – Simple Graphics using Turtle, Operations, 2D Shapes,	1 hour
3.2	Colors and RGB Systems, A case study.	1 hour
3.3	Image Processing – Basic image processing with inbuilt functions.	1 hour
3.4	Graphical User Interfaces – Event-driven programming	1 hour
3.5	Coding simple GUI-based programs : Windows, Labels, Displaying images,	1 hour
3.6	Coding simple GUI-based programs : Input text entry, Popup dialog boxes, Command buttons	1 hour
3.7	A case study - GUI	1 hour

Module 4: Object Oriented Programming		(7 hours)
4.1	Design with classes : Objects and Classes, Methods, Instance Variables	1 hour
4.2	Constructor, Accessors, and Mutators	1 hour
4.3	Structuring classes with Inheritance	1 hour
4.4	Polymorphism	1 hour
4.5	Abstract Classes	1 hour
4.6	Interfaces	1 hour
4.7	Exceptions : Handle a single exception, handle multiple exceptions	1 hour
Module 5: Data Processing		(9 hours)
5.1	The <i>os</i> and <i>sys</i> modules, NumPy : Basics, Creating arrays, Arithmetic, Slicing	1 hour
5.2	Matrix Operations, Random numbers.	1 hour
5.3	Matplotlib : Basic plot, Ticks, Labels, and Legends	1 hour
5.4	Working with CSV files	1 hour
5.5	Pandas : Reading, Manipulating	1 hour
5.6	Pandas : Processing Data and Visualize.	1 hour
5.7	Introduction to Microservices using Flask	1 hour
5.8	Introduction to Microservices using Flask	1 hour
5.9	Introduction to Microservices using Flask	1 hour

CST 372	DATA AND COMPUTER COMMUNICATION	Category	L	T	P	Credits	Year of Introduction
		PEC	2	1	0	3	2019

Preamble:

The purpose of this course is to prepare learners to understand the communication entities and the associated issues in data transmission. This course covers fundamental concepts of data transmission in digital and analog form, transmission media, concepts of encoding, multiplexing, spread spectrum and switching methods. This course helps the learner to gain insight into the important aspects of data communication and computer networking systems and enables to apply in practical applications.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Identify the characteristics of signals for analog and digital transmissions (Cognitive knowledge: Apply)
CO2	Identify the issues in data transmission (Cognitive knowledge: Apply)
CO3	Select transmission media based on characteristics and propagation modes (Cognitive knowledge: Apply)
CO4	Choose appropriate signal encoding techniques for a given scenario (Cognitive knowledge: Apply)
CO5	Illustrate multiplexing and spread spectrum technologies (Cognitive knowledge: Apply)
CO6	Use error detection, correction and switching techniques in data communication (Cognitive knowledge: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓
CO3	✓		✓									✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓
CO6	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40

Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Test : **25 marks**

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Data Transmission Basics)

Communication model - Simplex, Half duplex, Full duplex transmission. Periodic analog signals - Sine wave, Amplitude, Phase, Wavelength, Time and frequency domain, Bandwidth. Analog & digital data and signals. Transmission impairments - Attenuation, Delay distortion, Noise. Data rate limits - Noiseless channel, Nyquist bandwidth, Noisy channel, Shannon's capacity formula.

Module-2 (Transmission Media)

Guided transmission media - Twisted pair, Coaxial cable, Optical fiber. Unguided media - Radio waves, Terrestrial microwave, Satellite microwave, Infrared. Wireless propagation - Ground wave propagation, Sky wave propagation, Line-of-Sight (LoS) propagation.

Module-3 (Digital Transmission and Analog Transmission)

Digital data to digital signal – Non-Return-to-Zero (NRZ), Return-to-Zero (RZ), Multilevel binary, Biphasic. Analog data to digital signal - Sampling theorem, Pulse Code Modulation (PCM), Delta Modulation (DM). Digital data to analog signal - Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), Phase Shift Keying (PSK). Analog data to analog signal - Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM).

Module-4 (Multiplexing and Spread Spectrum)

Multiplexing - Frequency Division Multiplexing (FDM), Wavelength Division Multiplexing (WDM), Time Division Multiplexing (TDM), Characteristics, Synchronous TDM, Statistical TDM. Spread spectrum techniques - Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), Code Division Multiplexing, Code Division Multiple Access (CDMA).

Module-5 (Error Detection, Correction and Switching)

Digital data communication techniques - Asynchronous transmission, Synchronous transmission. Detecting and correcting errors - Types of errors, Parity check, Checksum, Cyclic Redundancy Check (CRC), Forward Error Correction (FEC), Hamming distance, Hamming code. Basic principles of switching - Circuit switching, Packet switching, Message switching.

Text Books

1. Forouzan B. A., Data Communications and Networking, 5/e, McGraw Hill, 2013.
2. William Stallings, Data and Computer Communication 9/e, Pearson Education, Inc.

References

1. Schiller J., Mobile Communications, 2/e, Pearson Education, 2009.
2. Curt M. White, Fundamentals of Networking and Communication 7/e, Cengage learning.

Course Level Assessment Questions**Course Outcome 1 (CO1):**

1. If the spectrum of a channel is between 3 MHz and 4 MHz and $SNR_{dB} = 24$ dB, calculate the Shannon capacity.
2. Assume that a periodic signal is composed of five sine waves with frequencies 200, 400, 600, 800 and 1000 Hz. Determine the bandwidth. Draw the spectrum assuming all components have a maximum amplitude of 5 V.

Course Outcome 2 (CO2):

1. Given a receiver with an effective noise temperature of 294 K and a bandwidth of 10 MHz. Find the thermal noise level at the receiver side in dBW.
2. The loss in a cable is usually defined in decibels per kilometer (dB/km). If the signal at the beginning of a cable with -0.3 dB/km has a power of 2 mW, determine the power of the signal at 5 km.

Course Outcome 3 (CO3):

1. Explain the reflective property of a parabolic antenna.
2. Two separate frequencies are used for uplink and downlink transmission in satellite communication. Give reason.

Course Outcome 4 (CO4):

1. Encode the data sequence 101011100 using Multilevel binary and Biphasic schemes.
2. Encode the data bits 00101101110001 using 2B1Q encoding scheme. Assume negative original level.

Course Outcome 5 (CO5):

1. The frequency spectrum of input signals will move to high frequency bands by the FDM process. Justify.
2. Four channels are multiplexed using TDM. If each channel sends 100 bytes/sec and we multiplex one byte per channel, determine the frame size, duration of a frame, frame rate and bit rate of link.

Course Outcome 6 (CO6):

1. Using the divisor polynomial $x^4 + x + 1$, determine the Cyclic Redundancy Check (CRC) for the dataword 10110100. Also, perform the checking at the receiver side.
2. How many redundancy bits are required to generate the Hamming code for a 7-bit data? Assuming even parity, generate the Hamming code for the 7-bit dataword 1001101. If the fifth bit from the left of the received codeword is changed to 0, can

this be detected? Give reasons for your answer.

Model Question Paper

QP CODE: _____

PAGES: 3

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 372

Course Name : Data and Computer Communication

Max Marks: 100

Duration: 3 Hours

PART A

(Answer All Questions. Each question carries 3 marks)

1. Define bandwidth. Find the lowest frequency, if a periodic signal has a bandwidth of 20 Hz and the highest frequency is 60 Hz. Draw the spectrum if the signal contains all frequencies of the same amplitude.
2. Assume that a TV picture is to be transmitted over a channel with 4.5 MHz bandwidth and a 35 dB Signal-to-Noise-Ratio. Find the capacity of the channel.
3. How does twisting affect the performance in a twisted pair cable?
4. Which wireless propagation method is suitable for satellite communication? Justify your answer.
5. Explain the two main distortions that can occur in a delta modulated waveform. How can it be avoided?
6. Illustrate the equivalent square wave pattern of the bit string 01001101 using Non-Return-to-Zero (NRZ) - Level and NRZ-Invert encoding schemes.
7. Apply Direct Sequence Spread Spectrum to the data 101 using the Barker sequence 10110111000. Show the encoding and decoding steps.
8. Compare synchronous and statistical time division multiplexing.
9. Find the minimum hamming distance for the following cases:
 - a) Detection of two errors

- b) Correction of two errors
- c) Detection of three errors

10. Find the parity bit for simple even parity check for the following.

- a) 1001010
- b) 0001100
- c) 1000000

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) With the help of suitable figures, distinguish between time and frequency domain representations. (4)

(b) Describe the different types of transmission impairments. (10)

OR

12. (a) Calculate the bandwidth, if a periodic signal is decomposed into 4 sine waves with frequencies 50 Hz, 100 Hz, 150 Hz and 200 Hz. Draw the spectrum, assuming all components having an amplitude in the range 6-12 V and all are multiples of two in the increasing order. (6)

(b) Distinguish between Nyquist bandwidth and Shannon capacity. Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with (i) Two signal levels (ii) Four signal levels. Determine the maximum bit rate in both cases. (8)

13. (a) For a parabolic reflective antenna operating at 12 GHz with a diameter of 2 m, calculate the effective area and the antenna gain. (6)

(b) List any four advantages and disadvantages of twisted pair, coaxial cable and fiber optic cable. (8)

OR

14. (a) Compare the features of terrestrial microwave and satellite microwave. (6)

(b) With the help of suitable diagrams, differentiate Multi-mode and Single-mode optical fibres. How are the rays propagated in Step-index and Graded-index Multi-mode fibres? (8)

15. (a) Distinguish between data rate and signal rate. (4)
- (b) How is polar encoding done? Encode the pattern 010011001110 using the two Biphase schemes. (10)

OR

16. (a) Show the equivalent analog sine wave pattern of the bit string 010011010 using Amplitude Shift Keying, Frequency Shift Keying and Phase Shift Keying. (4)
- (b) State Sampling theorem. Explain Pulse Code Modulation with suitable figures. (10)
17. (a) Four channels are multiplexed using Time Division Multiplexing. If each channel sends 100 bytes/sec and we multiplex one byte per channel, determine the frame size, duration of a frame, frame rate and bit rate of the link. (6)
- (b) Explain the working of Frequency Hopping Spread Spectrum with an example. (8)

OR

18. (a) Explain any three techniques by which the disparity in input data rate is handled by Time Division Multiplexing. Give examples. (4)
- (b) Suppose Alice and Bob are communicating using Code Division Multiple Access. Alice uses the code [+1 +1] and Bob uses the code [+1 -1]. Alice sends a data bit 0 and Bob sends a data bit 1. Show the data in the channel and how they can detect what the other person has sent. (10)
19. (a) Explain two-dimensional parity check with examples. (4)
- (b) Describe the need for a switch in a communication system. What are the different phases in circuit switching? (10)

OR

20. (a) Explain the virtual circuit approach of packet switching with a suitable example. (6)
- (b) Find the Hamming code for the data word 1011001. Assume odd parity. (8)

Teaching Plan

No	Contents	No.of Lecture Hrs (35 hrs)
Module-1 (Data Transmission Basics) (6 hrs)		
1.1	Introduction, Communication model, Simplex, Half duplex, Full duplex transmission, Periodic analog signals, Sine wave, Amplitude, Phase, Wavelength	1
1.2	Time and frequency domain, Bandwidth	1
1.3	Analog & digital data and signals	1
1.4	Transmission impairments, Attenuation, Delay distortion, Noise	1
1.5	Data rate limits, Noiseless channel, Nyquist bandwidth	1
1.6	Noisy channel, Shannon's capacity formula	1
Module-2 (Transmission Media) (6 hrs)		
2.1	Guided transmission media, Twisted pair, Coaxial cable	1
2.2	Optical fiber	1
2.3	Unguided media, Radio waves	1
2.4	Terrestrial microwave, Satellite microwave	1
2.5	Infrared	1
2.6	Wireless Propagation, Ground wave, Sky wave, Line-of-Sight (LoS) propagation	1
Module-3 (Digital Transmission and Analog Transmission) (8 hrs)		
3.1	Digital data to digital signal, Non-Return-to-Zero (NRZ), Return-to-Zero (RZ)	1
3.2	Multilevel binary and Biphasic	1
3.3	Analog data to digital signal, Sampling theorem, Pulse Code Modulation (PCM)	1

3.4	Delta Modulation (DM)	1
3.5	Digital data to analog signal, Amplitude Shift Keying (ASK)	1
3.6	Frequency Shift Keying (FSK), Phase Shift Keying (PSK)	1
3.7	Analog data to analog signal, Amplitude Modulation (AM)	1
3.8	Frequency Modulation (FM), Phase Modulation (PM)	1
Module-4 (Multiplexing and Spread Spectrum) (7 hrs)		
4.1	Multiplexing, Frequency Division Multiplexing (FDM), Wavelength Division Multiplexing (WDM)	1
4.2	Time Division Multiplexing (TDM), Characteristics, Synchronous TDM, Statistical TDM	1
4.3	Spread spectrum techniques, Direct Sequence Spread Spectrum (DSSS)	1
4.4	Frequency Hopping Spread Spectrum (FHSS)	1
4.5	Code Division Multiplexing	1
4.6	Code Division Multiple Access (CDMA) (Lecture 1)	1
4.7	CDMA (Lecture 2)	1
Module-5 (Error Detection, Correction and Switching) (8 hrs)		
5.1	Digital data communication techniques, Asynchronous & Synchronous transmission	1
5.2	Detecting and correcting errors, Types of errors, Parity check, Checksum	1
5.3	Cyclic Redundancy Check (CRC)	1
5.4	Forward Error Correction (FEC), Hamming distance	1
5.5	Hamming code	1
5.6	Basic principles of switching, Circuit switching	1
5.7	Packet switching	1
5.8	Message switching	1

APJ ABDUL KALAM
TECHNOLOGICAL
UNIVERSITY

SEMESTER VI

MINOR



CST 382	INTRODUCTION TO SOFTWARE TESTING	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This is a course in theoretical computer science that includes test cases for white-box, black-box, and grey-box approaches. This course describes the various techniques for test case design used to test software artifacts, including requirements, design, and code. The course includes different techniques for test case design based on graphs, programming language syntaxes and inputs. The course also covers symbolic execution using PEX tool.

Course Outcomes: After the completion of the course the student will be able to:-

CO1	List a range of different software testing techniques and be able to apply specific unit testing method to the projects using Junit. (Cognitive Knowledge Level: Understand)
CO2	Explain mutation testing method for a given piece of code to identify hidden defects that can't be detected using other testing methods. (Cognitive Knowledge Level: Understand)
CO3	Explain graph coverage criteria in terms of control flow graph and data flow graph for a given program. (Cognitive Knowledge Level: Understand)
CO4	Demonstrate the importance of black-box approaches in terms of domain and functional testing. (Cognitive Knowledge Level: Understand)
CO5	Illustrate the use of PEX tool with symbolic execution. (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO 9	PO10	PO11	PO12
CO1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>									<input checked="" type="checkbox"/>
CO2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

CO3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
CO4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>
CO5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	Test 1 (Marks)	Test 2 (Marks)	Marks
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment - Test : **25 marks**

Continuous Assessment - Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction to Software Testing)

Some Popular Errors – Ariane 5, Therac 25, Intel Pentium Bug. What is Software testing? Why should it be tested? Software Quality, Role of Testing. Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking. Software Testing Terminologies - Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria. Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing, Functional testing, Stress testing, Performance testing, Usability testing and Regression testing. Testing Methods - Black Box testing, White Box testing, Grey Box testing.

Module - 2 (Unit Testing)

Concept of Unit testing. Static Unit testing. Dynamic Unit testing - Control Flow testing, Data Flow testing, Domain testing, Functional Program testing. Mutation testing - Mutation and Mutants, Mutation operators, Mutation score. Junit - Framework for Unit testing. Case Study - Mutation testing using Junit and Muclipse.

Module - 3 (Unit Testing - White Box Approaches)

Overview of Graph Coverage Criteria. Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage, Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage. Data Flow Criteria - du paths, du pairs. Subsumption Relationships among Graph Coverage Criteria. Graph Coverage for Source Code - Control flow graphs for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program – Statistics. Graph Coverage for Design Elements - Call graphs and classes, Class inheritance testing: Coverage criteria, Coverage criteria on inheritance graph, Data flow at the design level, Inter-procedural DU pairs, Coupling du-pairs example. Example - Quadratic Root. Case Study - Graph Based testing using JUnit Framework.

Module - 4 (Unit Testing - Black Box Approaches)

Domain Testing / Input Space Partitioning - Partitions of a set. Input domain modelling - Interface-based approach, Functionality-based approach. Identifying values. Multiple partitions of the inputdomain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage. TriTyp example. Functional Testing - Functional Testing Concepts of Howden. Functional testing - Important Steps. Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Random Testing. Case Study - Black Box testing approaches using JUnit.

Module - 5 (Grey Box Testing Approaches)

Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages. Techniques of Grey Box Testing - Matrix Testing, Regression Testing, Orthogonal Array Testing or OAT, Pattern Testing. An Introduction to PEX - Parameterized Unit Testing, The Testing Problem. Symbolic Execution – Example, Symbolic execution tree. PEX application Case Study – PEX.

Text Books

1. Paul Ammann and Jeff Offutt, Introduction to Software Testing.
2. Kshirasagar Naik and Priyadarshi Tripathy, Software Testing And Quality Assurance: Theory And Practice.

Reference Materials

1. <https://www.csc.ncsu.edu/academics/undergrad/honors/thesis/muclipsebinder.pdf> - Muclipse tutorial.
2. King, James C, “Symbolic Execution and Program Testing”, Association for Computing Machinery, July 1976.
- 3.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): Explain the following types of testing methods with examples.

- (i) Black-box testing.
- (ii) White-box testing.
- (iii) Grey-box testing.

Course Outcome 2 (CO2): Define 12 mutants for the following method *power()* using effective mutation operators. Try to use each mutation operator at least once. Approximately, how many mutants do you think there would be, if all mutants for *power()* were created?

```
public static int power (int left, int right)
```

```
{
```

```
/**/
```

```
// Raises Left to the power of Right
```

```
// precondition : Right >= 0
```

```
// postcondition: Returns Left**Right
```

```
/**/
```

```
    intrslt;
```

```
    rslt = Left;
```

```

if (Right == 0)
{
    rslt = 1;
}
else
{
    for (int i = 2; i <= Right; i++)
        rslt = rslt * Left;
}
return (rslt);
}

```

Course Outcome 3 (CO3): Draw the control flow graph and data flow graph of given piece of code.

```

public static double ReturnAverage(int value[],int AS, int MIN, int MAX){
/*
Function: ReturnAverageComputes the averageof all those numbers in the input array in
the positive range [MIN, MAX]. The maximumsize of the array is AS. But, the array size
could be smaller than AS in which case the endof input is represented by -999.
*/
int i, ti, tv, sum;
doubleav;
i = 0; ti = 0; tv = 0; sum = 0;
while (ti< AS && value[i] != -999) {
ti++;
if (value[i] >= MIN && value[i] <= MAX) {
tv++;
sum = sum + value[i];
}
i++;
}
if (tv> 0)
av = (double)sum/tv;

```

```

else
av = (double) -999;
return (av);
}

```

Course Outcome 4 (CO4): Explain the following with examples.

1. Input domain modelling.
2. All Combinations Coverage (ACoC)
3. Each Choice Coverage (ECC)
4. Pair-wise Coverage
5. T-wise Coverage
6. Base Choice Coverage
7. Multiple Base Choices Coverage.

Course Outcome 5 (CO5): Draw the symbolic execution tree for the following program code and explain the symbolic execution of testme (α_1 , α_2).

```

1. int twice (int v) {
2.   return 2 * v;
3. }
4. void testme (int x, int y) {
5.   z = twice ( y);
6.   if ( z == x ){
7.     if ( x > y + 10)
8.       ERROR;
9.   }
10. }
11. int main() {
12.   x = sym input();
13.   y = sym input();
14.   testme ( x , y);
15.   return(0);
16. }

```

Model Question Paper

QP CODE:

PAGES: 4

Reg No: _____

Name : _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 382

Course Name: Introduction to Software Testing

Max.Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Explain the differences between Validation and Verification.
2. Explain the differences between Fault, Error, and Bug?
3. Define Ground string, Mutation score, and Mutants.
4. What are the functions of Test driver and Test stubs in dynamic unit testing?
5. Define Node coverage, Edge coverage and Prime path coverage in a control flow graph.
6. What are du paths and du pairs in a data flow graph?
7. Explain the two approaches in input domain modelling.
8. Explain the difference between Equivalence Class Partitioning and Boundary Value Analysis.
9. Briefly explain three techniques of Grey box testing.
10. Explain the concept of symbolic execution with the help of a toy example.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the following types of testing
(i) Black Box testing (ii) White Box testing (iii) Grey Box testing

(14)

(iv) Unit testing (v) Integration testing (vi) System testing (vii) Acceptance testing

OR

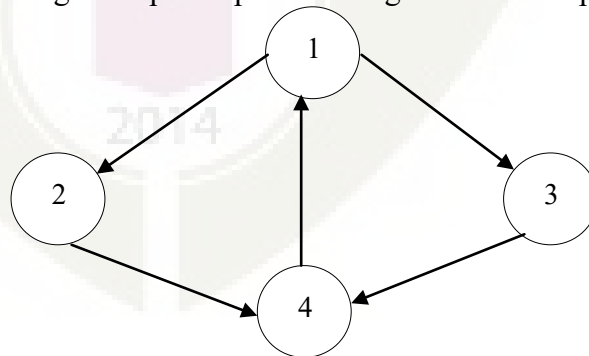
12. (a) Explain the following coverage criterias based on the code fragment given below. (i) Functional coverage (ii) Statement coverage (iii) Conditional coverage (iv) Branch coverage (8)

```
int foo (int x, int y){
    int z = 0;
    if ((x > 0) && (y > 0)){
        z = x;}
    return z;
}
```

- (b) Write positive and negative test cases for an ATM Machine? (6)
13. (a) Explain Dynamic unit test environment with a neat figure. (8)
- (b) Explain the major difference between control flow testing and data flow testing. (6)

OR

14. Explain seven types of mutation operators with neat examples. (14)
15. (a) Explain touring, side trips and detours with a neat example. (7)
- (b) Explain simple path coverage and prime path coverage with the help of CFG given below. (7)



OR

16. (a) Draw CFG fragment for

- (i) Simple *if* (ii) Simple *while* loop (iii) Simple *for* loop (7)
- (b) Explain the following concepts with examples. (7)
- (i) Call graph (ii) Inheritance graph (iii) Coupling du-pairs
17. (a) What are the four important steps in functional testing? (7)
- (b) Briefly explain input domain modelling approaches. (7)
- OR**
18. (a) Consider the triangle classification program with a specification: (6)
- The program reads floating values from the standard input. The three values A , B , and C are interpreted as representing the lengths of the sides of triangle. The program then prints a message to the standard output that states whether the triangle, if it can be formed, is scalene, isosceles, equilateral, or right angled. Determine the following for the above program:
- (i) For the boundary condition $A + B > C$ case (scalene triangle), identify test cases to verify the boundary.
- (ii) For the boundary condition $A = C$ case (isosceles triangle), identify test cases to verify the boundary.
- (iii) For the boundary condition $A = B = C$ case (equilateral triangle), identify test cases to verify the boundary.
- (b) Develop a decision table to generate test cases for this specification. (8)
19. (a) Explain the importance of grey box testing, its advantages and disadvantages. (9)
- (b) Explain the concept of symbolic execution tree. (5)
- OR**
20. (a) Consider the code fragment given below: - (7)
1. POWER: PROCEDURE(X, Y);
 2. Z ← 1;
 3. J ← 1;
 4. LAB: IF Y ≥ J THEN

5. DO; Z ← Z * X;
6. J ← J + 1;
7. GO TO LAB; END;
8. RETURN (Z) ;
9. END;

a) Explain Symbolic execution of POWER (α_1, α_2).

(b) Explain Execution tree for POWER (α_1, α_2) in the above code fragment. (7)

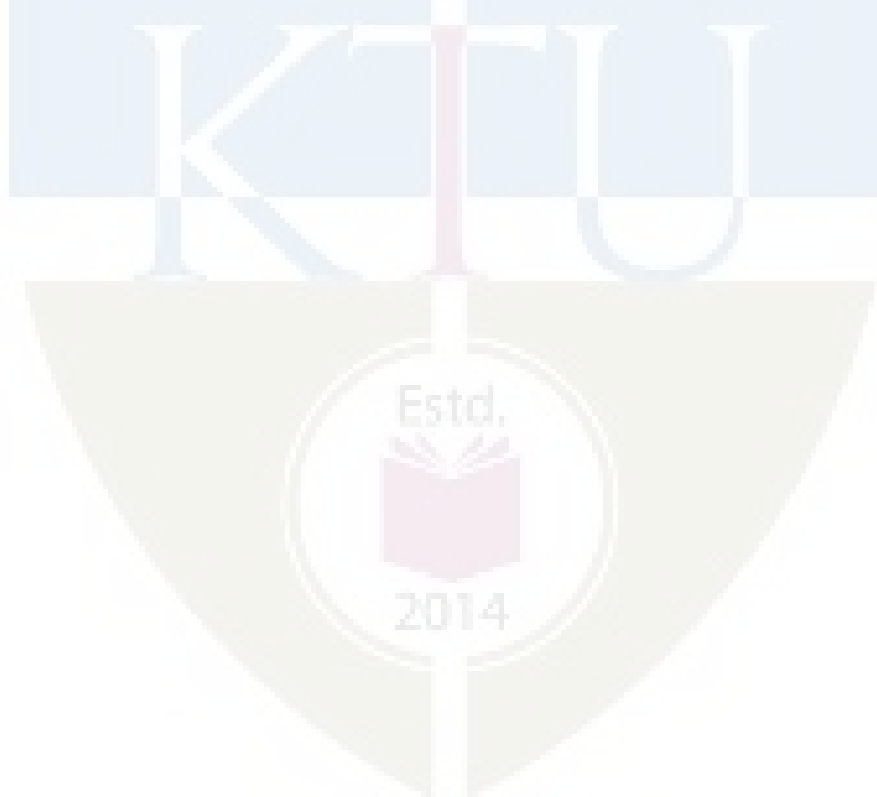
TEACHING PLAN

Index	Topics	No. of Hours (45)
Module 1 (Introduction to Software Testing) 9 Hours		
1.1	Some Popular Errors– Ariane 5, Therac 25, Intel Pentium Bug.	1 Hour
1.2	What is Software testing? Why should it be tested? Software Quality, Role of Testing.	1 Hour
1.3	Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking.	1 Hour
1.4	Software Testing Terminologies- Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria.	1 Hour
1.5	Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing	1 Hour
1.6	Functional testing, Stress testing	1 Hour
1.7	Performance testing, Usability testing and Regression testing.	1 Hour
1.8	Testing Methods - Black Box testing	1 Hour
1.9	Grey Box testing.	1 Hour
Module 2 (Unit testing) 8 Hours		

2.1	Concept of Unit testing.	1 Hour
2.2	Static Unit testing.	1 Hour
2.3	Dynamic Unit testing - Control Flow testing, Data Flow testing	1 Hour
2.4	Domain testing, Functional Program testing.	
2.5	Mutation testing - Mutation and Mutants, Mutation operators, Mutation score.	1 Hour
2.6	Junit - Framework for Unit testing.	1 Hour
2.7	Case Study - Mutation testing using Junit	1 Hour
2.8	Case Study - Mutation testing using Muclipse	1 Hour
Module 3 (Unit Testing:- White Box Approaches) 10 Hours		
3.1	Overview of Graph Coverage Criteria	1 Hour
3.2	Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage	1 Hour
3.3	Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage.	1 Hour
3.4	Data Flow Criteria - du paths, du pairs	1 Hour
3.5	Subsumption Relationships among Graph Coverage Criteria.	1 Hour
3.6	Graph Coverage for Source Code - Control Flow Graphs (CFG) for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program - Statistics	1 Hour
3.7	Graph Coverage for Design Elements - Call graphs and classes, Class inheritance testing: Coverage criteria, Coverage criteria on inheritance graph,	1 Hour

3.8	Data flow at the design level, Inter-procedural DU pairs, Coupling du-pairs example. Example - Quadratic Root	1 Hour
3.9	Case Study - Graph Based testing using JUnit Framework. (Lecture 1)	1 Hour
3.10	Case Study - Graph Based testing using JUnit Framework. (Lecture 2)	1 Hour
Module 4 (Unit Testing:- Black Box Approaches) 9 Hours		
4.1	Domain Testing / Input Space Partitioning - Partitions of a set.	1 Hour
4.2	Input domain modelling - Interface-based approach, Functionality-based approach.	1 Hour
4.3	Identifying values.	1 Hour
4.4	Multiple partitions of the input domain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage.	1 Hour
4.5	TriTyp example.	1 Hour
4.6	Functional Testing - Functional Testing Concepts of Howden. Important Steps.	1 Hour
4.7	Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis	1 Hour
4.8	Decision Tables, Random Testing.	1 Hour
4.9	Case Study - Black Box testing approaches using JUnit.	1 Hour
Module 5 (Grey Box Testing Approaches) 9 Hours		
5.1	Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages.	1 Hour
5.2	Techniques of Grey Box Testing - Matrix Testing, Regression Testing,	1 Hour

5.3	Orthogonal Array Testing or OAT, Pattern Testing.	1 Hour
5.4	An Introduction to Pex - Parameterized Unit Testing, The Testing Problem.	1 Hour
5.5	Symbolic Execution – Example, Symbolic execution tree.	1 Hour
5.6	PEX application.	1 hour
5.7	Case Study – PEX (Lecture 1)	1 Hour
5.8	Case Study – PEX (Lecture 2)	1 Hour
5.9	Case Study – PEX (Lecture 3)	1 Hour



CST 384	CONCEPTS IN DEEP LEARNING	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This course aims to introduce the learner to an overview of the concepts and algorithms involved in deep learning. Deep learning is a subfield of machine learning, a subfield of artificial intelligence. Basic concepts and application areas of machine learning, deep networks, convolutional neural network and recurrent neural network are covered here. This is a foundational program that will help students understand the capabilities, challenges, and consequences of deep learning and prepare them to participate in the development of leading-edge AI technology. They will be able to gain the knowledge needed to take a definitive step in the world of AI.

Prerequisite: Sound knowledge in Basics of linear algebra and probability theory.

CO1	Demonstrate basic concepts in machine learning.(Cognitive Knowledge Level: Understand)
CO2	Illustrate the validation process of machine learning models using hyper-parameters and validation sets. (Cognitive Knowledge Level: Understand)
CO3	Demonstrate the concept of the feed forward neural network and its training process. (Cognitive Knowledge Level: Apply)
CO4	Build CNN and Recurrent Neural Network (RNN) models for different use cases. (Cognitive Knowledge Level: Apply)
CO5	Use different neural network/deep learning models for practical applications. (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓	✓						✓
CO5	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

INTRODUCTION TO DEEP LEARNING

(General Instructions: Instructors are to introduce students to any one software platform and demonstrate the working of the algorithms in the syllabus using suitable use cases and public datasets to give a better understanding of the concepts discussed. Tutorial hour may be used for this purpose)

Module-1 (Introduction)

Key components - Data, models, objective functions, optimization algorithms, Learning algorithm. Supervised learning- regression, classification, tagging, web search, page ranking, recommender systems, sequence learning, Unsupervised learning, Reinforcement learning, Historical Trends in Deep Learning. Other Concepts - overfitting, underfitting, hyperparameters and validation sets, estimators, bias and variance.

Module- 2 (Optimization and Neural Networks)

Neural Networks –Perceptron, Gradient Descent solution for Perceptron, Multilayer perceptron, activation functions, architecture design, chain rule, back propagation, gradient based learning. Introduction to optimization– Gradient based optimization, linear least squares. Stochastic gradient descent, Building ML algorithms and challenges.

Module -3 (Convolutional Neural Network)

Convolutional Neural Networks – convolution operation, motivation, pooling, Structure of CNN, Convolution and Pooling as an infinitely strong prior, variants of convolution functions, structured outputs, data types, efficient convolution algorithms. Practical challenges of common deep learning architectures- early stopping, parameter sharing, dropout. Case study: AlexNet, VGG, ResNet.

Module- 4 (Recurrent Neural Network)

Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs.

Module-5 (Application Areas)

Applications – computer vision, speech recognition, natural language processing, common word embedding: continuous Bag-of-Words, Word2Vec, global vectors for word representation (GloVe). Research Areas – autoencoders, representation learning, boltzmann machines, deep belief networks.

Text Book

1. Ian Goodfellow, YoshuaBengio, Aaron Courville, Deep Learning, MIT Press 2015 ed.
2. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Dive into Deep Learning, August 2019.
3. Neural Networks and Deep Learning, Aggarwal, Charu C., c Springer International Publishing AG, part of Springer Nature 2018

Reference Books

1. Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks by Russell Reed, Robert J MarksII, A Bradford Book,2014
2. Practical Convolutional Neural Networks by MohitSewak, Md. Rezaul Karim, PradeepPujari, Packt Publishing 2018
3. Hands-On Deep Learning Algorithms with Python by SudharsanRavichandran, Packt Publishing 2019
4. Deep Learning with Python by Francois Chollet, Manning Publications Co., 2018

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

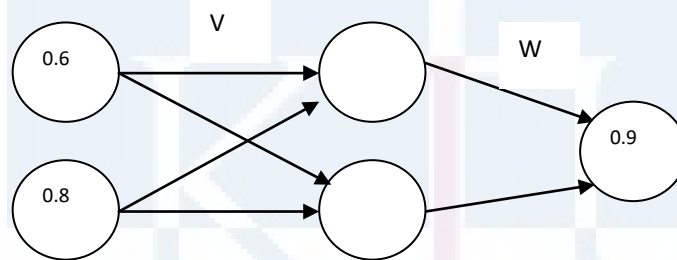
1. Compare regression and classification.
2. Define supervised learning? Distinguish between regression and classification.
3. Discuss the different learning approaches used in machine learning.

Course Outcome 2 (CO2):

1. What are hyperparameters? Why are they needed?
2. What issues are to be considered while selecting a model for applying machine learning in a given problem?

Course Outcome 3 (CO3):

1. Update the parameters V_{11} in the given MLP using back propagation with learning rate as 0.5 and activation function as sigmoid. Initial weights are given as $V_{11}=0.2$, $V_{12}=0.1$, $V_{21}=0.1$, $V_{22}=0.3$, $V_{11}=0.2$, $W_{11}=0.5$, $W_{21}=0.2$



2. Draw the architecture of a multi-layer perceptron.
3. Derive update rules for parameters in the multi-layer neural network through the gradient descent.

Course Outcome 4 (CO4):

1. Give two benefits of using convolutional layers instead of fully connected ones for visual tasks.
2. Suppose that a CNN was trained to classify images into different categories. It performed well on a validation set that was taken from the same source as the training set but not on a testing set. What could be the problem with the training of such a CNN? How will you ascertain the problem? How can those problems be solved?
3. Explain how the cell state is updated in the LSTM model from C_{t-1} to C_t
4. Show the steps involved in an LSTM to predict stock prices.

Course Outcome 5 (CO5):

1. Explain how the cell state is updated in the LSTM model from C_{t-1} to C_t
2. Show the steps involved in an LSTM to predict stock prices.
3. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.

Course Outcome 6 (CO6):

1. Development a deep learning solution for problems in the domain i) natural language processing or ii Computer vision (Assignment)
2. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.

Model Question Paper

QP CODE:

PAGES:4

Reg No: _____

Name: _____

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
SIXTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR**

Course Code: CST 384

Course Name: CONCEPTS IN DEEP LEARNING

Max. Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Distinguish between supervised learning and Reinforcement learning. Illustrate with an example.
2. Differentiate classification and regression.
3. Compare overfitting and underfitting. How it can affect model generalization.

4. Why does a single perceptron cannot simulate simple XOR function? Explain how this limitation is overcome?
5. Illustrate the strengths and weaknesses of convolutional neural networks.
6. Illustrate convolution and pooling operation with an example
7. How many parameters are there in AlexNet? Why the dataset size (1.2 million) is important for the success of AlexNet?
8. Explain your understanding of unfolding a recursive or recurrent computation into a computational graph.
9. Illustrate the use of deep learning concepts in Speech Recognition.
10. What is an autoencoder? Give one application of an autoencoder

(10x3=30
)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” What is your understanding of the terms task, performance and experience. Explain with two example
- (b) “How does bias and variance trade-off affect machine learning algorithms?

(10)

(4)

OR

12. (a) Illustrate the concepts of Web search, Page Ranking, Recommender systems with suitable examples.
- (b) List and discuss the different hyper parameters used in fine tuning the

(10)

(4)

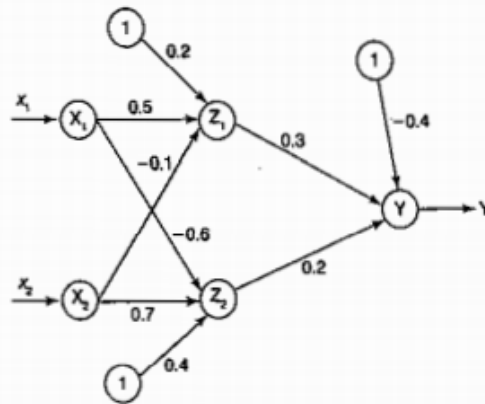
traditional machine learning models

13. (a) How multilayer neural networks learn and encode higher level features from input features. (7)

- (b) Explain gradient decent and delta rule? Why stochastic approximation to gradient descent is needed? (7)

OR

14. (a) Find the new weights for the network using backpropagation algorithm, the network is given with a input pattern[-1,1] and target output as +1, Use learning rate of $\alpha=0.3$ and bipolar sigmoid function. (7)



- (b) Write an algorithm for backpropagation which uses stochastic gradient descent method. Comment on the effect of adding momentum to the network. (7)

15. (a) Input to CNN architecture is a color image of size $112 \times 112 \times 3$. The first convolution layer comprises of 64 kernels of size 5×5 applied with a stride of 2 and padding 0. What will be the number of parameters? (5)

- (b) Let $X = [-1, 0, 3, 5]$ $W = [0.3, 0.5, 0.2, 0.1]$ be the the input of i^{th} layer of a neural network and to apply softmax function. What should be the output of it? (4)

- (c) Draw and explain the architecture of convolutional network (5)

OR

16. (a) Explain the concept behind i) Early stopping ii) dropout iii) weight decay (9)

- (b) How backpropagation is used to learn higher-order features in a convolutional Network? (5)
17. (a) Explain the working of RNN and discuss how backpropagation through time is used in recurrent networks. (8)
- (b) Describe the working of a long short term memory in RNNs. (6)
- OR**
18. (a) What is the vanishing gradient problem and exploding gradient problem? (8)
- (b) Why do RNNs have a tendency to suffer from exploding/vanishing gradient? How to overcome this challenge? (6)
19. (a) Explain any two word embedding techniques (8)
- (b) Explain the merits and demerits of using Auto encoders in Computer Vision. (6)
- OR**
20. (a) Illustrate the use of representation learning in object classification. (7)
- (b) Compare Boltzmann Machine with Deep Belief Network. (7)

Teaching Plan

CONCEPTS IN DEEP LEARNING (45 Hours)		
Module 1 : Introduction (9 hours)		
1.1	Key components - Data, models, objective functions, optimization algorithms. (TB2: Section 1.1-1.2)	1 hour

1.2	Learning algorithm (TB1: Section 5.1), Supervised learning- regression, classification (TB2: Section 1.3.1)	1 hour
1.3	tagging, web search, page ranking (TB2: Section 1.3.1)	1 hour
1.4	Recommender systems, Sequence learning, Unsupervised learning, Reinforcement learning(TB2: Section 1.3.2-1.3.4)	1 hour
1.5	Historical Trends in Deep Learning (TB1: Section 1.2).	1 hour
1.6	Concepts: over-fitting, under-fitting, hyperparameters and validation sets. (TB1: Section 5.2-5.3)	1 hour
1.7	Concepts: Estimators, bias and variance. (TB1: Section 5.4)	1 hour
1.8	Demonstrate the concepts of supervised learning algorithms using a suitable platform.	1 hour
1.9	Demonstrate the concepts of unsupervised using a suitable platform.	1 hour
Module 2 : Optimization and Neural Networks (9 hours)		
2.1	Perceptron, Stochastic Gradient descent, Gradient descent solution for perceptron (TB3: Section 1.1 - 1.2.1)	1 hour
2.2	Multilayer perceptron (TB3: Section 1.2.2), (TB1: Section 6.1,6.3)	1 hour
2.3	Activation functions- Sigmoid, tanh, Softmax, ReLU, leaky ReLU (TB3: Section 1.2.1.3 - 1.2.1.5)	1 hour
2.4	Architecture design (TB1: Section 6.4, TB3: Section 1.6)	1 hour
2.5	Chain rule, back propagation (TB3: Section 1.3)	1 hour

2.6	Gradient based learning (TB1: Section 6.2)	1 hour
2.7	Gradient based optimization (TB1: Section 4.3)	1 hour
2.8	Linear least squares using a suitable platform. (TB1: Section 4.5)	1 hour
2.9	Building ML Algorithms and Challenges (TB3: 1.4, TB1: 5.10-5.11)	1 hour
Module 3 :Convolution Neural Network (10 hours)		
3.1	Convolution operation, Motivation, pooling (TB1:Section 9.1-9.3)	1 hour
3.2	Structure of CNN (TB3: Section 8.2)	1 hour
3.3	Convolution and Pooling as an infinitely strong prior (TB1: Section 9.4)	1 hour
3.4	Variants of convolution functions – multilayer convolutional network, tensors, kernel flipping, downsampling, strides and zero padding. (TB1: Section 9.5)	1 hour
3.5	Variants of convolution functions - unshared convolutions, tiled convolution, training different networks. (TB1: Section 9.5)	1 hour
3.6	Structured outputs, data types (TB1: Section 9.6-9.7)	1 hour
3.7	Efficient convolution algorithms. (TB1: Section 9.8,9.10)	1 hour
3.8	Practical challenges of common deep learning architectures- early Stopping (TB3: 4.6)	1 hour
3.9	Practical challenges of common deep learning architectures- parameter sharing, drop-out (TB3: Section 4.9, 4.5.4)	1 hour
3.10	Case Study: AlexNet,VGG, ResNet. (TB3: Section 8.4.1-8.4.3,8.4.5)	1 hour

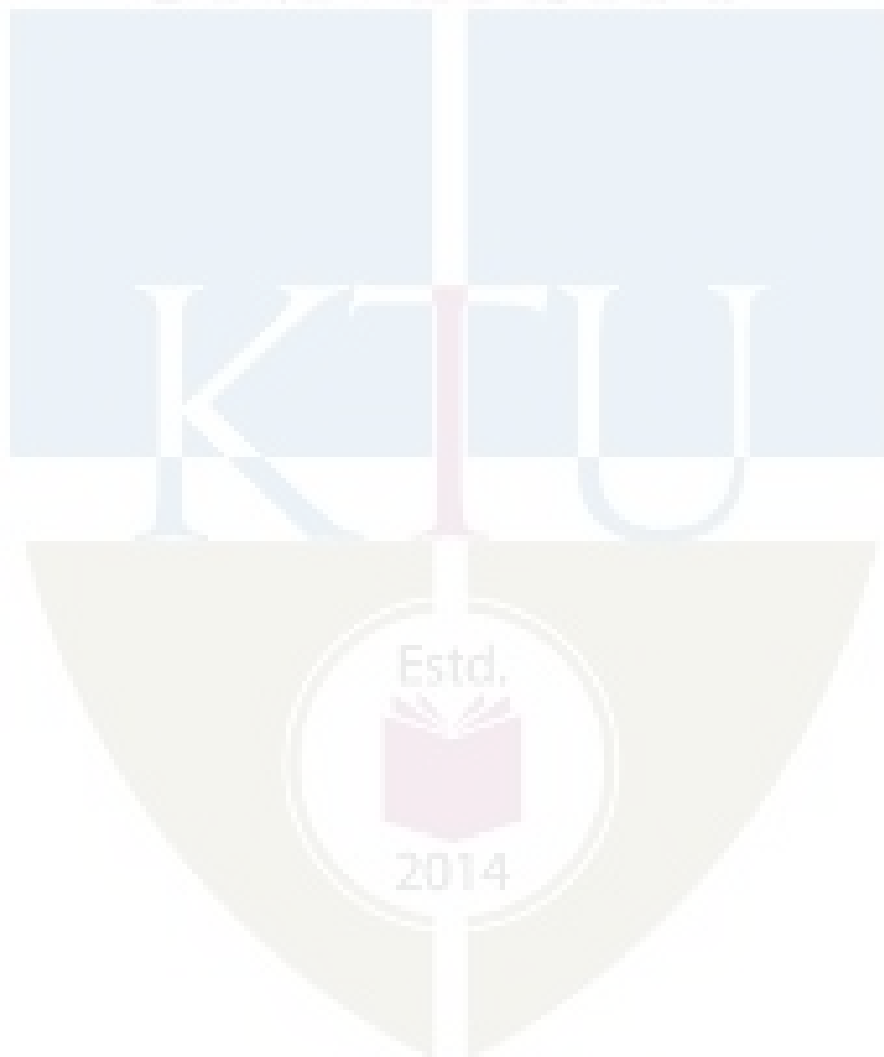
Module 4 :Recurrent Neural Network (8 hours)

4.1	Computational graphs (TB1: Section 10.1)	1 hour
4.2	RNN (TB1: Section 10.2-10.3)	1 hour
4.3	Encoder – decoder sequence to sequence architectures. (TB1: Section 10.4)	1 hour
4.4	Deep recurrent networks (TB1: Section 10.5)	1 hour
4.5	Recursive neural networks , Modern RNNs, (TB1: Section 10.6, 10.10)	1 hour
4.6	LSTM and GRU (TB1: Section 10.10, TB3: Section 7.5-7.6)	1 hour
4.7	Practical use cases for RNNs. (TB1: Section 11.1-11.4)	1 hour
4.8	Demonstrate the concepts of RNN using a suitable platform.	1 hour

Module 5 : Applications and Research (9 hours)

5.1	Computer vision. (TB1: Section 12.2)	1 hour
5.2	Speech recognition. (TB1: Section 12.3)	1 hour
5.3	Natural language processing. (TB1: Section 12.4)	1 hour
5.4	Common Word Embedding -: Continuous Bag-of-Words, Word2Vec (TB3: Section 2.6)	1 hour
5.5	Common Word Embedding -: Global Vectors for Word Representation(GloVe) (TB3: Section 2.9.1- Pennigton 2014)	1 hour
5.6	Brief introduction on current research areas- Autoencoders, Representation learning. (TB3: Section 4.10)	1 hour

5.7	Brief introduction on current research areas- representation learning. (TB3: Section 9.3)	1 hour
5.8	Brief introduction on current research areas- Boltzmann Machines, Deep belief networks. (TB1: Section 20.1, TB3 Section 6.3)	1 hour
5.9	Brief introduction on current research areas- Deep belief networks. (TB1: Section 20.3)	1 hour



CST 386	WIRELESS NETWORKS AND IoT APPLICATIONS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This course equips the learners with fundamental wireless technologies for the Internet of Things(IoT) and the IoT ecosystem. It covers the underlying concepts in wireless networks, communication mechanisms, protocols, hardware, software, and the cloud platforms for IoT. The students will be able to design smart IoT applications for real world problems..

Prerequisite: Sound knowledge in Data Communication, Computer Networks and Programming in C

Course Outcomes: After the completion of the course the students will be able to

CO1	Recognize wireless technologies required for IoT ecosystem (Cognitive Knowledge Level : Understand)
CO2	Perceive the concept of IoT and M2M architecture, IoT examples, and Data Management in IoT (Cognitive Knowledge Level :Apply)
CO3	Outline the hardware components used in IoT including Sensors, Actuators and development boards (Cognitive Knowledge Level : understand)
CO4	Explain the software components of IoT (Cognitive Knowledge Level :Understand)
CO5	Demonstrate the protocols used in IoT and build IoT Programs (Cognitive Knowledge Level : Apply)
CO6	Build IoT-based smart real-time applications such as Smart Healthcare, Smart Agriculture, Smart Environment and Smart Home (Cognitive Knowledge Level : Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												

CO3												
CO4												
CO5												
CO6												

Abstract POs Defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and teamwork
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Blooms Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Percentage)	Test 2 (Percentage)	
Remember	30	30	30
Understand	50	40	40
Apply	20	30	30

Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 Hours

Continuous Internal Evaluation Pattern:

Attendance	10 marks
Continuous Assessment Tests	25 marks
Continuous Assessment Assignment	15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First Internal Examination shall be preferably conducted after completing the first half of the syllabus, and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer anyone. Each question can have a maximum 2 subdivisions and carries 14 marks.

Syllabus

Module- 1 (Introduction to IoT and wireless technologies required for IoT)

Internet of Things, Role of Things and the Internet, Wireless IoT. Wireless Networks - Network Topologies, Types of Networks. Role of Wireless Standards in IoT. Protocol Stack - OSI Model, TCP/IP Model, IEEE 802 Reference Model, Protocols for Wireless IoT. Bluetooth - Transceiver, Frequency Channels, Typical Range, Access and Spread Spectrum, Modulation and Data Rate, Error Correction and Detection, Network Topology. ITU G.9959, Zwave, IEEE 802.15.4, Zigbee Specification, Thread, WiFi, 6LowPAN, IPv6, LoRaWAN.

Module- 2 (IoT architecture, Data and Device management)

Internet of Things - IoT Architectural View, Technology Behind IoT - Server End Technology, Sources of Internet of Things, M2M Communication. IoT Application Areas. IoT Examples. IoT Data Management - Device Management Gateways. Design Principles for Web Connectivity - Web Communication Protocols for Connected Devices, Web Connectivity for Connected Devices using Gateways. Internet Connectivity Principles – Internet Connectivity, Internet based communication, IP addressing in the IoT.

Module- 3 (Data Acquiring and Enabling Technologies)

Data Acquiring and Storage for IoT Services- Organization of Data, Big data, Acquiring Methods, Management Techniques, Analytics, Storage Technologies. Cloud Computing for Data storage - IoT Cloud based Services using Xively, Nimbits, and Other Platforms. Sensor Technologies for IoT Devices - Sensor Technology, Participatory Sensing, Industrial IoT and Automotive IoT, Actuators for Various Devices, Sensor Data Communication Protocols, Wireless Sensor network Technology

Module-4 (Prototyping the Embedded Devices for IoT)

Embedded Computing Basics, Embedded Hardware Unit. Embedded Platforms for Prototyping - Arduino, Intel Galileo, Intel Edison, Raspberry Pi, BeagleBone, mBed. Prototyping and Designing the Software for IoT Applications- Introduction, Prototyping Embedded Device Software- Programming using Arduino, Programming for an Arduino Controlled Traffic Control Lights at a Road Junction, Basic Arduino Programs to Blink LED, Find the Distance using Ultrasonic Sensor, Estimate Room Temperature, Measuring Soil Moisture Level

Module 5 (Business Models and Case Studies)

Business Models and Processes using IoT. Value Creation in the Internet of Things. Cloud PaaS- Xively, Nimbits, IBM Bluemix, CISCO IoT, AWS IoT, TCS Connected AWS Platform, Case studies- Smart Home, Smart Environment, Smart healthcare, Smart agriculture

Text Books

1. Daniel Chew, “Wireless Internet of Things -A Guide to the lower layers”, IEEE Standards and Association, IEEE Press, Wiley
2. Rajkamal, “Internet of Things : Architecture and Design Principles”, McGraw Hill (India) Private Limited.

References

1. ArshadeepBahga, Vijay Madiseti, “Internet of Things: A hands-on approach”, University Press, 2015 (First edition)
2. Dieter Uckelmann, Mark Harrison, Michahelles Florian (Ed.), Architecting the internet of things, Springer, 2011
3. Dr. Ovidiu Vermesan, Dr. Peter Friess, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers, 2013
4. Simon Monk, “Programming Arduino: Getting Started with Sketches”, McGraw Hill Publications

Sample Course Level Assessment Questions**Course Outcome 1 (CO1):**

1. Compare Bluetooth and Bluetooth LE power classes
2. Demonstrate Zigbee Specification Protocol Stack

Course Outcome 2 (CO2):

1. What are the major components of IOT system? Briefly explain each
2. Correlate M2M architectural Levels with IOT architectural Levels

Course Outcome 3 (CO3):

1. Describe the use of GPIO pins ?
2. What are actuators ? Mention the roles of actuators in IoT systems

Course Outcome 4(CO4):

1. Identify the role of HBase in Hadoop File System
2. Differentiate Edge computing and Distributed computing
3. Illustrate open protocols, tools and frameworks generally used in M2M

Course Outcome 5(CO5):

1. What do you mean by Arduino sketches?
2. Write an Arduino program to blink LED

Course Outcome 6(CO6):

1. How IoT technology helps TELEMEDICINE in India?
2. How soil moisture can be detected in Smart Agriculture?

Model Question Paper

QP CODE:

PAGES :2

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 386

Course Name: WIRELESS NETWORKS AND IoT APPLICATIONS

Max.Marks:100

Duration: 3 Hours

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Illustrate Role of *things* and *internet* in IoT
2. What is Bluetooth? Explain the range and frequency channels of Bluetooth?
3. List any three the features of Constrained Application Protocol (COAP).
4. Compare Raspberry Pi and BeagleBoard boards.
5. Identify the role of HBase in Hadoop File System.
6. Differentiate Edge computing and Distributed computing.
7. Give an example of Raspberry Pi applications for Industrial IoT.
8. What are the on-board functional units in Intel Galileo?
9. Interpret the concept of value creation in IoT.

10. Explain the use of PaaS in IoT Smart applications with any three examples.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Compare various Network topologies used in Wireless Networks. (8)

(b) Describe the following wireless technologies on i) *Zigbee* ii) *WiFi*
iii) *Thread*. (6)

OR

12. (a) Explain protocol stacks used in wireless networks for IoT applications. (8)

(b) Illustrate the Architectural design of LoRaWAN. (6)

13. (a) Define M2M. Explain M2M architecture. Correlate M2M architectural levels with IoT architectural levels. (8)

(b) Compare SOAP and REST protocols. (6)

OR

14. (a) Summarize different Online Transactions and Processing techniques. (8)

(b) Identify the functions of Device-Management Gateway . (6)

15. (a) Define actuators ? Describe the roles of actuators in IoT systems. (8)

(b) Explain the usage contexts of analog sensors and digital sensors. (6)

OR

16. (a) How data collection, storage & computing services done using Nimbits? (10)

(b) List any four features of Xively. (4)

17. (a) What do you mean by Arduino sketches? (4)
 (b) Write an Arduino program to blink LED (10)

OR

18. (a) Demonstrate an example of Raspberry Pi applications for Industrial IoT. (10)
 (b) Compare the features of Arduino-R3 and Arduino Yun boards. (4)
19. (a) Explain various tasks of a smart irrigation monitoring service. (8)
 (b) Demonstrate the tasks of Soil-Moisture monitoring service. (6)

OR

20. (a) a) Mr. Kiran Mathew has been a chronic diabetic patient for the past few years. He was under regular check up at the hospital every two weeks. All of a sudden the pandemic like COVID-19 arises in the country and the government issues a lockdown for a period of two months. Illustrate how Mr. Kiran can be monitored by the health care worker using intelligent healthcare techniques. (10)
 (b) Mention any four sensors used in smart healthcare (4)

TEACHING PLAN

No	Contents	No of Lecture Hrs(45)
Module – 1 (Introduction to IoT and wireless technologies required for IoT) (8 hrs) (TB-1, Chapter 1...)		
1.1	Internet Of Things, Role of things and internet ,Wireless IoT	1
1.2	Wireless Networks- Network Topologies-Types of Networks,Role of	1

	Wireless standards in IoT	
1.3	Protocol Stack-OSI Model- TCP/IP Model-IEEE 802 reference model	1
1.4	Protocols for Wireless IoT-Bluetooth-Transceiver, Frequency Channels-Typical Range, Access and Spread Spectrum, Modulation and Data Rate	1
1.5	Error Correction and Detection-Network Topology.	1
1.6	ITU G.9959, Zwave, IEEE 802.15.4, Zigbee Specification	1
1.7	Thread, Wifi, 6LowPAN, IPv6	1
1.8	LoRaWAN	1
Module- 2 (IOT architecture, Data and Device management) (9hrs)		
2.1	Internet of Things -IoT Architectural view	1
2.2	Technology Behind IOT-Server End Technology,Sources of Internet of Things	1
2.3	M2M Communication.	1
2.4	IoT Application Areas. IOT Examples.	1
2.5	IoT Data Management, Device Management Gateways.	1
2.6	Design Principles for Web Connectivity	1
2.7	Web communication protocols for connected devices,	1
2.8	Web connectivity for connected devices using Gateways.	1
2.9	Internet connectivity Principles – Internet Connectivity, Internet based communication, IP addressing in the IoT.	1
Module- 3 (Data Acquiring and Enabling Technologies (8 hrs)		
3.1	Data acquiring and storage for IoT devices- Organization of Data, Big data	1
3.2	Acquiring methods, management techniques, Analytics, Storage technologies.	1
3.3	Cloud computing for Data storage-IoT Cloud based services using Xively,	1

	Nimbits, and other platforms.	
3.4	Cloud computing-Nimbits	1
3.5	Sensor Technologies for IoT Devices-Sensor Technology, Participatory sensing	1
3.6	Industrial IoT and Automotive IoT	1
3.7	Actuators for various devices, Sensor data communication protocols	1
3.8	Wireless Sensor network Technology	1
Module 4(Prototyping the Embedded Devices for IoT)(9hrs)		
4.1	Introduction, Embedded Computing Basics, Embedded Hardware Unit.	1
4.2	Embedded Platforms for Prototyping-Arduino, Intel Galileo	1
4.3	Intel Edison, Raspberry Pi, BeagleBone, mBed	1
4.4	Prototyping and designing the software for IoT applications-Introduction, Prototyping embedded device software	1
4.5	Prototyping and designing the software for IoT applications-Introduction, Prototyping embedded device software	1
4.6	Programming concepts in Arduino	1
4.7	Programming for an arduino controlled traffic control lights at a road junction	1
4.8	Basic Arduino programs to blink LED, Find the distance using ultrasonic sensor	1
4.9	Estimate room temperature, Measuring soil moisture level	1
Module 5 (higher level protocols and case studies)(9 hrs)		
5.1	Business Models and Processes using IOT, Value creation in the Internet of Things.	1

5.2	Xively, Nimbits, IBM Bluemix	1
5.3	CISCO IoT, AWS IoT, TCS Connected AWS Platform	1
5.4	Case Study- Smart Environment	1
5.5	Case Study- Smart Environment	1
5.6	Case study Smart Home	1
5.7	Case study Smart Home	1
5.8	Case study Smart healthcare (Lecture I)	1
5.9	Case study Smart healthcare (Lecture II)	1
5.10	Case study -Smart agriculture (Lecture I)	1
5.11	Case study -Smart agriculture (Lecture II)	1

APJ ABDUL KALAM
TECHNOLOGICAL
UNIVERSITY

SEMESTER VI

HONOURS



CST 394	NETWORK SECURITY	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

The purpose of this course is to create a better understanding of the network security concepts. This course covers network security standards, email security services, web security mechanisms, firewalls and wireless security mechanisms. This course helps the learner to gain insight into the key aspects of secure network communication and enables to apply in real-life scenarios.

Prerequisite: A sound background in Number Theory and Cryptographic Algorithms.

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Identify the key aspects of security, intrusion detection systems and digital signature schemes (Cognitive Knowledge Level: Apply)
CO2	Explain the security standards used in network communication (Cognitive Knowledge Level: Understand)
CO3	Identify the mechanisms in email security services (Cognitive Knowledge Level: Apply)
CO4	Summarize the protocols used to provide web security (Cognitive Knowledge Level: Understand)
CO5	Explain the fundamental concepts of wireless network security and firewalls (Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓		✓						✓
CO4	✓	✓	✓	✓	✓	✓						✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Network Security Basics)

Introduction to network security - Security requirements, Challenges of security, Network security model. Malicious programs – Worms, Viruses, Trojans, Spyware, Adware. Intrusion Detection Systems (IDS) - Uses, Techniques. Digital signatures - ElGamal, Schnorr, Digital Signature Standard (DSS).

Module – 2 (Network Security Standards)

Kerberos v4 – Configuration, Authentication, Encryption, Message formats. Kerberos v5 – Cryptographic algorithms, Message formats. Public Key Infrastructure (PKI) – Trust models, Revocation. Real-time communication security – Perfect Forward Secrecy (PFS), Denial-of-Service protection, Endpoint identifier hiding, Live partner reassurance. Internet Protocol Security (IPSec) - Authentication Header (AH), Encapsulating Security Payload (ESP), Internet Key Exchange (IKE) phases.

Module – 3 (Email Security)

Introduction to email security - Security services for email, Establishing keys, Privacy, Authentication, Message integrity, Non-repudiation. Privacy Enhanced Mail (PEM) – Encryption, Source authentication and integrity protection, Message formats. Secure/Multipurpose Internet Mail Extensions (S/MIME) – Messages, Differences from PEM. Pretty Good Privacy (PGP) - Encoding, Certificate and key revocation, Anomalies, Object formats.

Module – 4 (Web Security)

Introduction to web security - Web security considerations, Threats. Secure Sockets Layer (SSL) – Architecture, Protocols, Transport Layer Security (TLS) – Differences from SSL. Hypertext Transfer Protocol Secure (HTTPS) – Connection initiation, Closure. Secure Shell (SSH) – Transport layer protocol, User authentication protocol, Connection protocol.

Module – 5 (Wireless Network Security and Firewalls)

IEEE 802.11 Wireless LAN - Network components, Architectural model, Services. IEEE 802.11i wireless LAN security - Services, Phases of operation. Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), WPA2, Wireless Application Protocol (WAP) – Services, Protocol architecture. Firewalls – Need for firewalls, Packet filters, Circuit-level firewalls, Application layer firewalls.

Text Books

1. C. Kaufman, R. Perlman and M. Speciner, “Network Security: Private Communication in a Public World”, 2/e, PHI.
2. William Stallings, “Cryptography and Network Security Principles and Practice”, 5/e, Pearson

Education Asia.

References

1. Behrouz A. Forouzan, Debdeep Mukhopadhyay, “Cryptography and Network Security”, 3/e, Tata McGraw Hill.
2. Tyler Wrightson, “Wireless Network Security A Beginner’s Guide”, 2012, Tata McGraw Hill.
3. William Stallings, “Network Security Essentials: Applications and Standards”, 4/e, Prentice Hall.
4. Schiller J., Mobile Communications, 2/e, Pearson Education.
5. Roberta Bragg et. al., “Network Security: The Complete Reference”, Tata McGraw Hill.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Using the Schnorr digital signature scheme, let $q = 83$, $p = 997$ and $d = 23$. Find values for e_1 and e_2 .
2. The Digital Signature Algorithm (DSA) specifies that if the signature generation process results in a value of zero, a new value of k should be generated and the signature should be recalculated. Give reason.

Course Outcome 2 (CO2):

1. In Kerberos v4, the authenticator field is not of security benefit when asking the Key Distribution Center (KDC) for a ticket for Bob, but useful when logging in as Bob. Give reasons for your answer.
2. How does the stateless cookie protocol provide clogging protection?

Course Outcome 3 (CO3):

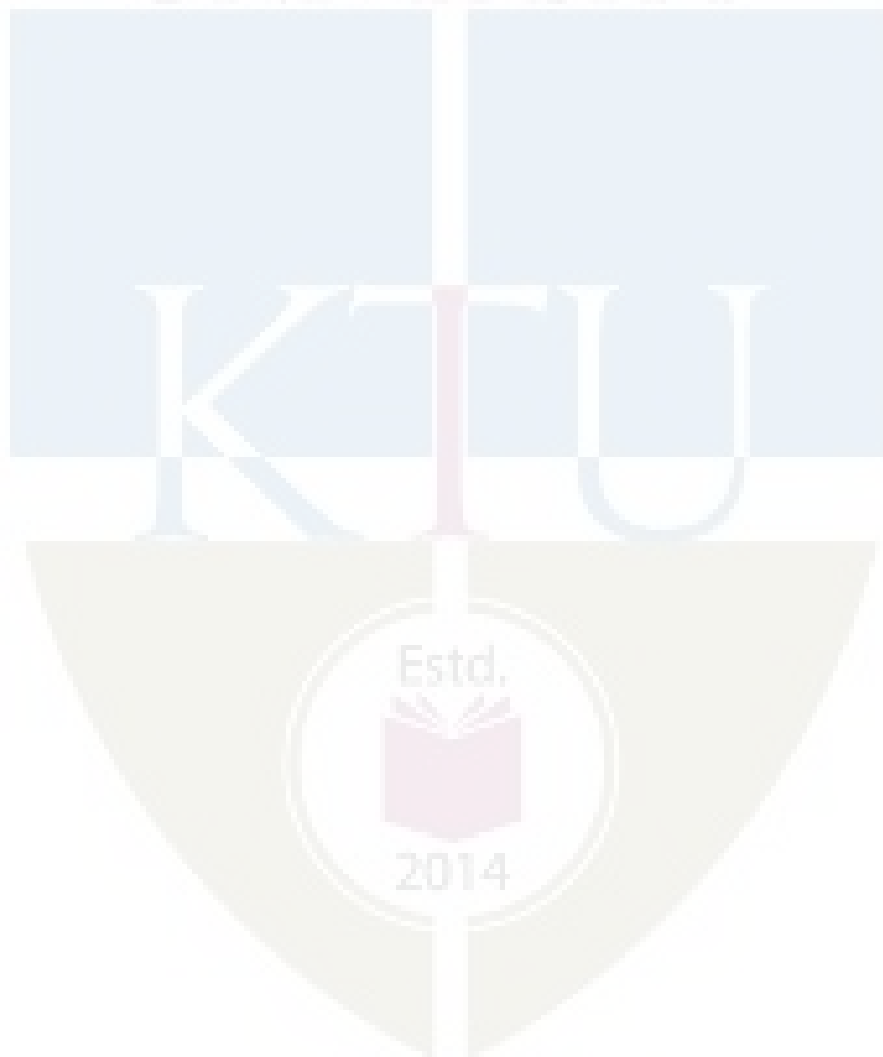
1. If Alice is sending an ENCRYPTED message, she first signs the message digest with her private key and then encrypts the message digest with the pre-message secret key. Why this last encryption was considered necessary for encrypted messages and not for MIC-CLEAR or MIC-ONLY?
2. Which security services are considered desirable in the following cases? (i) Sending a purchase order (ii) Sending a ransom note. (iii) Sending a mission description to security officials.
3. Explain the security mechanism used in Gmail communication.

Course Outcome 4 (CO4):

1. Is it possible in SSL for the receiver to reorder SSL record blocks that arrive out of order? If so, how it can be done? If not, why?
2. Describe any five web security threats, their consequences and countermeasures.

Course Outcome 5 (CO5):

1. Explain the security areas addressed by IEEE 802.11i.
2. Describe the advantages and disadvantages of application layer firewalls.



Model Question Paper

QP CODE:

Reg. No: _____

Name: _____

PAGES : 3

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
SIXTH SEMESTER B.TECH. DEGREE (HONORS) EXAMINATION, MONTH & YEAR
Course Code: CST 394

Course Name: Network Security

Max.Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Distinguish between signature-based and anomaly-based intrusion detection techniques.
2. A trusted third party is considered as a main component in a network security model. Why?
3. How is endpoint identifier hiding achieved in real-time communication?
4. Show how encryption is used to provide privacy and integrity in Kerberos v5.
5. End-to-end privacy is essential for e-mail security. How is this achieved?
6. List the four steps for preparing an EnvelopedData MIME entity.
7. Show the operation of a Secure Sockets Layer (SSL) Record protocol.
8. For Secure Shell (SSH) packets, what is the advantage of not including the MAC in the scope of packet encryption?
9. List the three security services provided by IEEE 802.11i.
10. Define the terms Access Point, Basic Service Set, Extended Service Set.

(10x3=30)

Part B**(Answer any one question from each module. Each question carries 14 Marks)**

11. (a) Using the ElGamal scheme, let $p = 881$ and $d = 700$, find values for e_1 and e_2 . Choose $r = 17$. Find the value of S_1 and S_2 if $M = 400$. (8)
- (b) Explain the requirements and challenges of network security. (6)

OR

12. (a) In ElGamal, Schnorr and DSS, what happens if an attacker can find the value of random secret key used by the signer? Also, what happens if a user uses the same value of random secret key to sign two messages? Explain your answer for each scheme separately. (8)
- (b) Explain the network security model with the help of a neat diagram. (6)

13. (a) Alice wishes to log into Bob's workstation remotely. List the steps involved in this communication if Kerberos v4 is used. (7)
- (b) How does Diffie-Hellman technique provide perfect forward secrecy using signature keys? (7)

OR

14. (a) Explain the algorithm for Message Authentication Code (MAC) calculation and verification in Kerberos v5 rsa-md5-des. (8)
- (b) Compare the aggressive mode and main mode of Phase 1 Internet Key Exchange (IKE). (6)

15. (a) Describe the different methods by which authentication of source is performed in email communication. (7)
- (b) Explain the Signed data and Clear-signed data functions provided by S/MIME. (7)

OR

16. (a) Explain the advantages of Pretty Good Privacy (PGP) over Privacy Enhanced Mail (PEM). (7)

- (b) Define non-repudiation. Describe the different ways by which it is implemented in email communication. (7)
17. (a) Describe the significance of pseudo-random function of Transport Layer Security. (7)
- (b) Explain the four different phases of Secure Sockets Layer (SSL) Handshake Protocol. (7)
- OR**
18. (a) Describe how connection initiation and connection closure is done in Hyper Text Transfer Protocol Secure (HTTPS). (7)
- (b) Illustrate the sequence of events in Secure Shell (SSH) transport layer protocol packet exchanges. (7)
19. (a) Explain the Discovery phase and Authentication phase of IEEE 802.11i operation. (7)
- (b) Why are firewalls needed? Compare the features of packet filters and circuit level firewalls. (7)
- OR**
20. (a) Explain the two authentication methods used in Wired Equivalent Privacy (WEP). (7)
- (b) Describe the three transaction classes provided by Wireless Transaction Protocol. (7)

Teaching Plan

No	Contents	No of Lecture Hrs
Module - 1 (Network Security Basics) (7 hrs)		
1.1	Security requirements, Challenges of security	1
1.2	Network security model	1
1.3	Worms, Viruses, Trojans, Spyware, Adware	1
1.4	Intrusion Detection Systems (IDS) uses, Techniques	1
1.5	ElGamal digital signature	1
1.6	Schnorr digital signature	1
1.7	Digital Signature Standard (DSS)	1
Module - 2 (Network Security Standards) (12 hrs)		
2.1	Kerberos v4 configuration, Authentication	1
2.2	Kerberos v4 encryption	1
2.3	Kerberos v4 message formats	1
2.4	Kerberos v5 cryptographic algorithms – rsa-md5-des, des-mac, des-mac-k	1
2.5	Kerberos v5 cryptographic algorithms - rsa-md4-des, rsa-md4-des-k, Encryption for privacy and integrity	1
2.6	Kerberos v5 message formats	1
2.7	Public Key Infrastructure (PKI) trust models	1
2.8	PKI revocation	1
2.9	Perfect Forward Secrecy (PFS), Denial-of-Service protection	1
2.10	Endpoint identifier hiding, Live partner reassurance	1
2.11	Internet Protocol Security (IPSec) Authentication Header (AH), Encapsulating Security Payload (ESP)	1

2.12	Internet Key Exchange (IKE) phases	1
Module - 3 (Email Security) (9 hrs)		
3.1	Security services for email, Establishing keys, Privacy	1
3.2	Authentication, Message integrity, Non-repudiation	1
3.3	Privacy Enhanced Mail (PEM) encryption, Source authentication	1
3.4	PEM integrity protection, Message formats (Lecture 1)	1
3.5	PEM message formats (Lecture 2)	1
3.6	Secure/Multipurpose Internet Mail Extensions (S/MIME) – Messages, Differences from PEM	1
3.7	Pretty Good Privacy (PGP) encoding, Certificate and key revocation, Anomalies	1
3.8	PGP Object formats (Lecture 1)	1
3.9	PGP Object formats (Lecture 2)	1
Module – 4 (Web Security)(9 hrs)		
4.1	Web security considerations, Threats, Secure Sockets Layer (SSL) architecture	1
4.2	SSL protocols (Lecture 1)	1
4.3	SSL protocols (Lecture 2)	1
4.4	Transport Layer Security (TLS) differences from SSL (Lecture 1)	1
4.5	TLS differences from SSL (Lecture 2)	1
4.6	Hypertext Transfer Protocol Secure (HTTPS) connection initiation, Closure	1
4.7	Secure Shell (SSH) transport layer protocol	1
4.8	SSH user authentication protocol	1
4.9	SSH connection protocol	1

Module - 5 (Wireless Security and Firewalls) (8 hrs)		
5.1	IEEE 802.11 Wireless LAN network components, Architectural model, Services	1
5.2	IEEE 802.11i wireless LAN security services, Phases of operation (Lecture 1)	1
5.3	IEEE 802.11i phases of operation (Lecture 2)	1
5.4	Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), WPA2	1
5.5	Wireless Application Protocol (WAP) services, Protocol architecture (Lecture 1)	1
5.6	WAP protocol architecture (Lecture 2)	1
5.7	Need for firewalls, Packet filters	1
5.8	Circuit-level firewalls, Application layer firewalls	1

CST 396	ADVANCED TOPICS IN MACHINE LEARNING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This course enables the learners to understand the advanced concepts and algorithms in machine learning. The course covers the standard and most popular supervised learning algorithms such as linear regression, logistic regression, decision trees, Bayesian learning and the naive Bayes algorithm, basic clustering algorithms, auto encoders, sampling methods and PAC learning. This course helps the students to provide machine learning based solutions to real world problems.

Prerequisite: Basic understanding of probability theory, linear algebra, multivariate calculus and multivariate probability theory.

CO1	Illustrate the concepts of regression and classification techniques (Cognitive Knowledge Level: Apply)
CO2	Demonstrate various unsupervised learning techniques (Cognitive Knowledge Level: Apply)
CO3	Choose suitable model parameters for different machine learning techniques and to evaluate a model performance (Cognitive Knowledge Level: Apply)
CO4	Explain the framework of PAC learning, basic concepts of VC dimension and non-uniform learnability (Cognitive Knowledge Level: Understand)
CO5	Construct Bayesian models for data and apply computational techniques to draw inferences (Cognitive Knowledge Level: Apply)
CO6	Illustrate the concepts of sampling algorithms, auto encoder, generative adversarial networks (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓	✓	✓						✓
CO2	✓	✓	✓	✓	✓	✓						✓
CO3	✓	✓	✓	✓	✓	✓						✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓	✓							✓
CO6	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum 2 sub-divisions and carry 14 marks.

Syllabus

Module -1 (Supervised Learning)

Overview of machine learning - supervised, semi-supervised, unsupervised learning, reinforcement learning Regression algorithms: least squares linear regression, gradient descent, closed form, normal equations, regularization techniques (LASSO, RIDGE), polynomial regression. Discriminative Methods - Logistic Regression, Decision Tree Learning. Generative Methods - Naive Bayes Classifier, Gaussian Discriminant Analysis (GDA).

Module -2 (Unsupervised Learning)

Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitional clustering, K-medoids clustering, Gaussian mixture models: Expectation Maximization (EM) algorithm for Gaussian mixture model.

Module -3 (Practical aspects in machine learning)

Classification Performance measures - Precision, Recall, Accuracy, F-Measure, ROC, AUC, generalisation and overfitting, cross-validation, bias-variance tradeoff, error estimation, parameter and model selection. Ensemble Methods - Bagging, Boosting, Adaboost, Random Forests.

Module -4 (Statistical Learning Theory)

Models of learnability- learning in the limit, probably approximately correct (PAC) learning. Sample complexity- quantifying the number of examples needed to PAC learn, Computational complexity of training, Sample complexity for finite hypothesis spaces, PAC results for learning conjunctions, Sample complexity for infinite hypothesis spaces, Vapnik-Chervonenkis(VC) dimension.

Module -5 (Advanced Machine Learning Topics)

Graphical models - Bayesian belief networks, Markov random fields(MRFs), Inference on chains and factor graphs, inference on clique trees. Monte Carlo methods – Basic sampling algorithms, rejection sampling, importance sampling, Markov chain Monte Carlo(MCMC), Gibbs sampling. Variational methods. Auto Encoder, Variational AutoEncoder, Generative Adversarial Networks

Textbook

1. Christopher M. Bishop. Pattern recognition and machine learning. Springer 2006.
2. Ethem Alpaydin, Introduction to Machine Learning, 2nd edition, MIT Press 2010.
3. Mohammed J. Zaki and Wagner Meira, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, First South Asia edition, 2016.
4. Ian Goodfellow, Yoshua Bengio and Aaron Courville. Deep Learning. MIT Press 2016.
5. Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar. Foundations of Machine Learning. Second edition. MIT Press 2018.
6. Tom Mitchell. Machine Learning. McGraw Hill 1997.
7. Richard O. Duda, Peter E . Hart, David G. Stork. Pattern classification, Second Edition. Wiley.
8. Jiawei Han, Micheline Kamber, Jian Pei. Data Mining Concepts and Techniques, Third Edition. Morgan Kaufmann.
9. David Foster. Generative Deep Learning - Teaching Machines to Paint, Write, Compose, and Play. O'Reilly Media, Inc., June 2019.

Reference Books

1. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. MIT Press 2012
2. Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning. MIT Press 2005.

Sample Course Level Assessment Questions

Course Outcome1 (CO1):

1. Consider a naive Bayes classifier with 3 boolean input variables, X_1 , X_2 and X_3 , and one boolean output, Y . How many parameters must be estimated to train such a naive Bayes classifier? How many parameters would have to be estimated to learn the above classifier if we do not make the naive Bayes conditional independence assumption?
2. Describe the ID3 algorithm. Is the order of attributes identical in all branches of the decision tree?
3. Explain the difference between (batch) gradient descent and stochastic gradient descent. Give an example of when you might prefer one over the other.
4. Suppose that you are asked to perform linear regression to learn the function that outputs y , given the D -dimensional input x . You are given N independent data points, and that all the D attributes are linearly independent. Assuming that D is around 100, would you prefer the closed form solution or gradient descent to estimate the regressor?
5. Suppose you have a three class problem where class label $y \in \{0, 1, 2\}$ and each training example X has 3 binary attributes $X_1, X_2, X_3 \in \{0, 1\}$. How many parameters (probability distribution) do you need to know to classify an example using the Naive Bayes classifier?

Course Outcome 2(CO2):

1. Which similarity measure could be used to compare feature vectors of two images? Justify your answer.
2. Illustrate the strength and weakness of k-means algorithm.
3. Suppose you want to cluster the eight points shown below using k-means

	A_1	A_2
x_1	2	10
x_2	2	5
x_3	8	4
x_4	5	8
x_5	7	5
x_6	6	4
x_7	1	2
x_8	4	9

Assume that $k = 3$ and that initially the points are assigned to clusters as follows:

$C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_4, x_5, x_6\}$, $C_3 = \{x_7, x_8\}$. Apply the k -means algorithm until convergence, using the Manhattan distance.

4. Cluster the following eight points representing locations into three clusters: $A_1(2, 10)$, $A_2(2, 5)$, $A_3(8, 4)$, $A_4(5, 8)$, $A_5(7, 5)$, $A_6(6, 4)$, $A_7(1, 2)$, $A_8(4, 9)$.

Initial cluster centers are: $A_1(2, 10)$, $A_4(5, 8)$ and $A_7(1, 2)$.

The distance function between two points $a = (x_1, y_1)$ and $b = (x_2, y_2)$ is defined as $D(a, b) = |x_2 - x_1| + |y_2 - y_1|$

Use k -Means Algorithm to find the three cluster centers after the second iteration.

Course Outcome 3(CO3):

1. What is ensemble learning? Can ensemble learning using linear classifiers learn classification of linearly non-separable sets?
2. Describe boosting. What is the relation between boosting and ensemble learning?
3. Classifier A attains 100% accuracy on the training set and 70% accuracy on the test set. Classifier B attains 70% accuracy on the training set and 75% accuracy on the test set. Which one is a better classifier. Justify your answer.
4. What are ROC space and ROC curve in machine learning? In ROC space, which points correspond to perfect prediction, always positive prediction and always negative prediction? Why?
5. Suppose there are three classifiers A,B and C. The (FPR, TPR) measures of the three classifiers are as follows – A (0, 1), B (1, 1) , C (1,0.5). Which can be considered as a perfect classifier? Justify your answer.

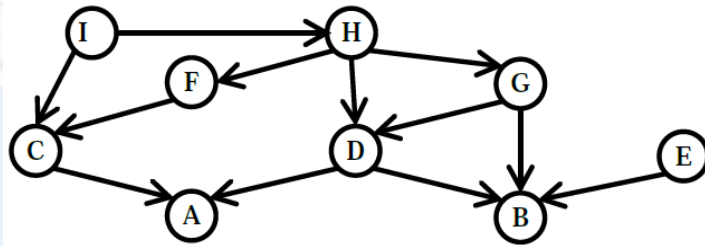
Course Outcome 4(CO4): .

1. A monotone conjunction is a conjunction of the variables such that no variable is negated. Show that the concept class of monotone conjunction is probably approximately correct (PAC)-learnable.
2. Consider a Boolean classification problem with n binary variables and a hypothesis space H , where each hypothesis is a decision tree of depth 2, using only two variables. How many training examples, m suffice to assure that with probability at least 0.99, any consistent learner using H will output a hypothesis with true error at most 0.05
3. Show that the concept class C containing the set of all boolean functions on n variable is not PAC-learnable.

4. What is the Vapnik-Chervonenkis(VC)-dimension of a circle centered at the origin.
5. A hypothesis space that has a high VC dimension is good, bad, or neither? Explain in terms of both (a) richness or expressive power of the hypotheses, and (b) sample complexity.

Course Outcome 5(CO5):

1. Write down the factored conditional probability expression that corresponds to the graphical Bayesian Network shown below.



2. How do we learn the conditional probability tables(CPT) in Bayesian networks if information about some variables is missing? How are these variables called?

Course Outcome 6 (CO6):

1. Derive an algorithm using the inverse transform method to generate a random sample from the exponential distribution.
2. Explain the pros and cons of importance sampling versus rejection sampling.
3. Sketch the core idea of the Monte Carlo method. What is a sample? What is a direct sampling method? Why can't it be used directly to do any inference? What is rejection sampling? What is its major disadvantage?
4. Generative Adversarial Networks(GANs) include a generator and a discriminator. Sketch a basic GAN using those elements, a source of real images, and a source of randomness.
5. The word “adversarial” in the acronym for GANs suggests a two-player game. What are the two players, and what are their respective goals?

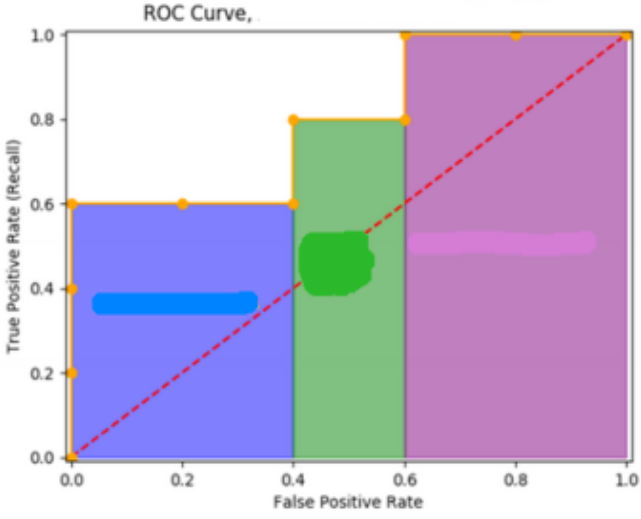
Model Question Paper**QP CODE:****Reg No:** _____**Name:** _____**PAGES : 5****APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY****SIXTH SEMESTER B.TECH DEGREE EXAMINATION (HONORS), MONTH & YEAR****Course Code: CST 396****Course Name: Advanced Topics in Machine Learning****Max.Marks:100****Duration: 3 Hours****PART A****Answer All Questions. Each Question Carries 3 Marks**

1.	Suppose you have a dataset with $m = 1000000$ examples and $n = 200000$ features for each example. You want to use multivariate linear regression to fit the parameters to our data. Should you prefer gradient descent or the normal equation? Justify your answer.	
2.	Define Information gain? How is that different from Gain ratio? Give the advantage of using Gain ratio measure?	
3.	What is cluster analysis? Identify two applications where cluster analysis can be applied to multimedia data?	
4.	Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8): (i) Compute the Euclidean distance between the two objects. (ii) Compute the Manhattan distance between the two objects.	
5.	Suppose there are three classifiers A,B and C. The (FPR, TPR) measures of the three classifiers are as follows – A (0, 1), B (1, 1) , C (1,0.5). Which can be considered as a perfect classifier? Justify your answer.	
6.	How Bias-Variance Tradeoff affects machine learning algorithms?	
7.	For a particular learning task, if the requirement of error parameter ϵ changes from 0.1 to 0.01. How many more samples will be required for probably approximately correct(PAC) learning?	

8.	Suppose we have a hypothesis set that labels all points inside an interval $[a, b]$ as class 1. Find its Vapnik-Chervonenkis(VC)- dimension?											
9.	Given a density function $f(x)$, the rejection sampling is a method that can generate data points from the density function f . List the three steps to generate a random sample from f using rejection sampling.											
10.	How does the variational auto-encoder(VAE) architecture allow it to generate new data points, compared to auto-encoder, which cannot generate new data points?	(10x3=30)										
Part B (Answer any one question from each module. Each question carries 14 Marks)												
11.	<p>(a) Consider the hypothesis for the linear regression $h_{\theta}(x) = \theta_0 + \theta_1 x$, and the cost function $J(\theta_0, \theta_1) = 1/2m \sum_{i=1}^{m} (h_{\theta}(x^{(i)}) - y^{(i)})^2$ where m is the number of training examples. Given the following set of training examples.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>x</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>2</td> </tr> <tr> <td>1</td> <td>2</td> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>4</td> <td>3</td> </tr> </tbody> </table> <p>Answer the following questions :</p> <p>1) Find the value of $h_{\theta}(2)$ if $\theta_0 = 0$ and $\theta_1 = 1.5$</p> <p>2) Find the value of $J(0,1)$</p> <p>3) Suppose the value of $J(\theta_0, \theta_1) = 0$. What can be inferred from this.</p>	x	y	3	2	1	2	0	1	4	3	(5)
x	y											
3	2											
1	2											
0	1											
4	3											
	(b) Write a gradient descent algorithm for multivariate regression? Compare the gradient and analytical solution to the multivariate regression?	(9)										
OR												
12.	(a) Consider the collection of training samples (S) in the Figure given below. Drug is the target attribute which describes the Drug suggested for each patient. Find the value of the following . i) Gain(S, Sex) ii) Gain (S, Cholesterol)	(9)										

		<table border="1"> <thead> <tr> <th>Patient ID</th> <th>Age</th> <th>Sex</th> <th>BP</th> <th>Cholesterol</th> <th>Drug</th> </tr> </thead> <tbody> <tr><td>p1</td><td>Young</td><td>F</td><td>High</td><td>Normal</td><td>Drug A</td></tr> <tr><td>p2</td><td>Young</td><td>F</td><td>High</td><td>High</td><td>Drug A</td></tr> <tr><td>p3</td><td>Middle-age</td><td>F</td><td>Hiigh</td><td>Normal</td><td>Drug B</td></tr> <tr><td>p4</td><td>Senior</td><td>F</td><td>Normal</td><td>Normal</td><td>Drug B</td></tr> <tr><td>p5</td><td>Senior</td><td>M</td><td>Low</td><td>Normal</td><td>Drug B</td></tr> <tr><td>p6</td><td>Senior</td><td>M</td><td>Low</td><td>High</td><td>Drug A</td></tr> <tr><td>p7</td><td>Middle-age</td><td>M</td><td>Low</td><td>High</td><td>Drug B</td></tr> <tr><td>p8</td><td>Young</td><td>F</td><td>Normal</td><td>Normal</td><td>Drug A</td></tr> <tr><td>p9</td><td>Young</td><td>M</td><td>Low</td><td>Normal</td><td>Drug B</td></tr> <tr><td>p10</td><td>Senior</td><td>M</td><td>Normal</td><td>Normal</td><td>Drug B</td></tr> <tr><td>p11</td><td>Young</td><td>M</td><td>Normal</td><td>High</td><td>Drug B</td></tr> <tr><td>p12</td><td>Middle-age</td><td>F</td><td>Normal</td><td>High</td><td>Drug B</td></tr> <tr><td>p13</td><td>Middle-age</td><td>M</td><td>High</td><td>Normal</td><td>Drug B</td></tr> <tr><td>p14</td><td>Senior</td><td>F</td><td>Normal</td><td>High</td><td>Drug A</td></tr> </tbody> </table>	Patient ID	Age	Sex	BP	Cholesterol	Drug	p1	Young	F	High	Normal	Drug A	p2	Young	F	High	High	Drug A	p3	Middle-age	F	Hiigh	Normal	Drug B	p4	Senior	F	Normal	Normal	Drug B	p5	Senior	M	Low	Normal	Drug B	p6	Senior	M	Low	High	Drug A	p7	Middle-age	M	Low	High	Drug B	p8	Young	F	Normal	Normal	Drug A	p9	Young	M	Low	Normal	Drug B	p10	Senior	M	Normal	Normal	Drug B	p11	Young	M	Normal	High	Drug B	p12	Middle-age	F	Normal	High	Drug B	p13	Middle-age	M	High	Normal	Drug B	p14	Senior	F	Normal	High	Drug A	
Patient ID	Age	Sex	BP	Cholesterol	Drug																																																																																								
p1	Young	F	High	Normal	Drug A																																																																																								
p2	Young	F	High	High	Drug A																																																																																								
p3	Middle-age	F	Hiigh	Normal	Drug B																																																																																								
p4	Senior	F	Normal	Normal	Drug B																																																																																								
p5	Senior	M	Low	Normal	Drug B																																																																																								
p6	Senior	M	Low	High	Drug A																																																																																								
p7	Middle-age	M	Low	High	Drug B																																																																																								
p8	Young	F	Normal	Normal	Drug A																																																																																								
p9	Young	M	Low	Normal	Drug B																																																																																								
p10	Senior	M	Normal	Normal	Drug B																																																																																								
p11	Young	M	Normal	High	Drug B																																																																																								
p12	Middle-age	F	Normal	High	Drug B																																																																																								
p13	Middle-age	M	High	Normal	Drug B																																																																																								
p14	Senior	F	Normal	High	Drug A																																																																																								
	(b)	Explain how LASSO regression helps to reduce the overfitting problem?	(5)																																																																																										
13.	(a)	<p>Suppose that we have the following data:</p> <table border="1"> <thead> <tr> <th><i>a</i></th> <th><i>b</i></th> <th><i>c</i></th> <th><i>d</i></th> <th><i>e</i></th> <th><i>f</i></th> <th><i>g</i></th> <th><i>h</i></th> <th><i>i</i></th> <th><i>j</i></th> </tr> </thead> <tbody> <tr> <td>(2,0)</td> <td>(1,2)</td> <td>(2,2)</td> <td>(3,2)</td> <td>(2,3)</td> <td>(3,3)</td> <td>(2,4)</td> <td>(3,4)</td> <td>(4,4)</td> <td>(3,5)</td> </tr> </tbody> </table> <p>Identify the cluster by applying the k-means algorithm, with $k = 2$. Try using initial cluster centers as far apart as possible.</p>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	(2,0)	(1,2)	(2,2)	(3,2)	(2,3)	(3,3)	(2,4)	(3,4)	(4,4)	(3,5)	(9)																																																																						
<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>																																																																																				
(2,0)	(1,2)	(2,2)	(3,2)	(2,3)	(3,3)	(2,4)	(3,4)	(4,4)	(3,5)																																																																																				
	(b)	Describe EM algorithm for Gaussian mixtures.	(5)																																																																																										
OR																																																																																													
14.	(a)	Illustrate the strength and weakness of k-means in comparison with the k-medoids algorithm.	(4)																																																																																										
	(b)	Suppose that we have the following data . Use single linkage Agglomerative clustering to identify the clusters. Draw the Dendogram. Use Euclidean distance measure	(10)																																																																																										

		<table border="1"> <thead> <tr> <th></th> <th>X</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>P1</td> <td>0.4</td> <td>0.53</td> </tr> <tr> <td>P2</td> <td>0.22</td> <td>0.38</td> </tr> <tr> <td>P3</td> <td>0.35</td> <td>0.32</td> </tr> <tr> <td>P4</td> <td>0.26</td> <td>0.19</td> </tr> <tr> <td>P5</td> <td>0.08</td> <td>0.41</td> </tr> <tr> <td>P6</td> <td>0.45</td> <td>0.30</td> </tr> </tbody> </table>		X	Y	P1	0.4	0.53	P2	0.22	0.38	P3	0.35	0.32	P4	0.26	0.19	P5	0.08	0.41	P6	0.45	0.30	
	X	Y																						
P1	0.4	0.53																						
P2	0.22	0.38																						
P3	0.35	0.32																						
P4	0.26	0.19																						
P5	0.08	0.41																						
P6	0.45	0.30																						
15.	(a)	Define Precision, Recall, Accuracy and F-measure?	(4)																					
	(b)	What does it mean for a classifier to have a high precision but low recall?	(3)																					
	(c)	Fill in the missing values in the accompanying three class confusion matrix. Given that model accuracy is 72% and classification error for class 2 is 20%. Find also the precision and recall for class 1	(7)																					
		<table border="1"> <thead> <tr> <th colspan="2" rowspan="2"></th> <th colspan="3">Predicted</th> </tr> <tr> <th>Class 1</th> <th>Class 2</th> <th>Class 3</th> </tr> </thead> <tbody> <tr> <th rowspan="3">Actual</th> <th>Class 1</th> <td>14</td> <td>2</td> <td>5</td> </tr> <tr> <th>Class 2</th> <td>?(X)</td> <td>40</td> <td>2</td> </tr> <tr> <th>Class 3</th> <td>1</td> <td>?(Y)</td> <td>18</td> </tr> </tbody> </table>			Predicted			Class 1	Class 2	Class 3	Actual	Class 1	14	2	5	Class 2	?(X)	40	2	Class 3	1	?(Y)	18	
		Predicted																						
		Class 1	Class 2	Class 3																				
Actual	Class 1	14	2	5																				
	Class 2	?(X)	40	2																				
	Class 3	1	?(Y)	18																				
OR																								

16.	(a)	What are ROC space and ROC curve in machine learning? In ROC space, which points correspond to perfect prediction, always positive prediction and always negative prediction? Why?	(4)
	(b)	<p>Given the following ROC Curve? Find the AUC?</p>  <p>The ROC Curve plot shows the True Positive Rate (Recall) on the y-axis and the False Positive Rate on the x-axis, both ranging from 0.0 to 1.0. A red dashed diagonal line represents a random classifier. The ROC curve is a step function defined by the following points: (0, 0), (0.2, 0.6), (0.4, 0.8), (0.6, 1.0), and (1.0, 1.0). The area under the curve is shaded in three colors: blue for the region where FPR < 0.4 and TPR < 0.6; green for the region where 0.4 < FPR < 0.6 and 0.4 < TPR < 0.8; and purple for the region where FPR > 0.6 and TPR > 0.4.</p>	(3)
	(c)	How does random forest classifier work? Why is a random forest better than a decision tree?	(7)
17.	(a)	Show that the concept class C_n of the conjunction of boolean literals up to n variables is probably approximately correct(PAC)-learnable.	(8)
	(b)	Explain the concept of Vapnik-Chervonenkis (VC) dimension using shattering. How the number of training examples required to train the model is related to the VC dimension and what is its relation with training and test errors.	(6)
OR			
18.	(a)	Consider a Boolean classification problem with n binary variables and a hypothesis space H , where each hypothesis is a decision tree of depth 1. How many training examples, m suffice to assure that with probability at least 0.99, any consistent learner using H will output a hypothesis with true error at most 0.05?	(7)
	(b)	Prove that $VC(H) \leq \log_2 H $, where H is a hypothesis space. ($ H $ denotes the	(7)

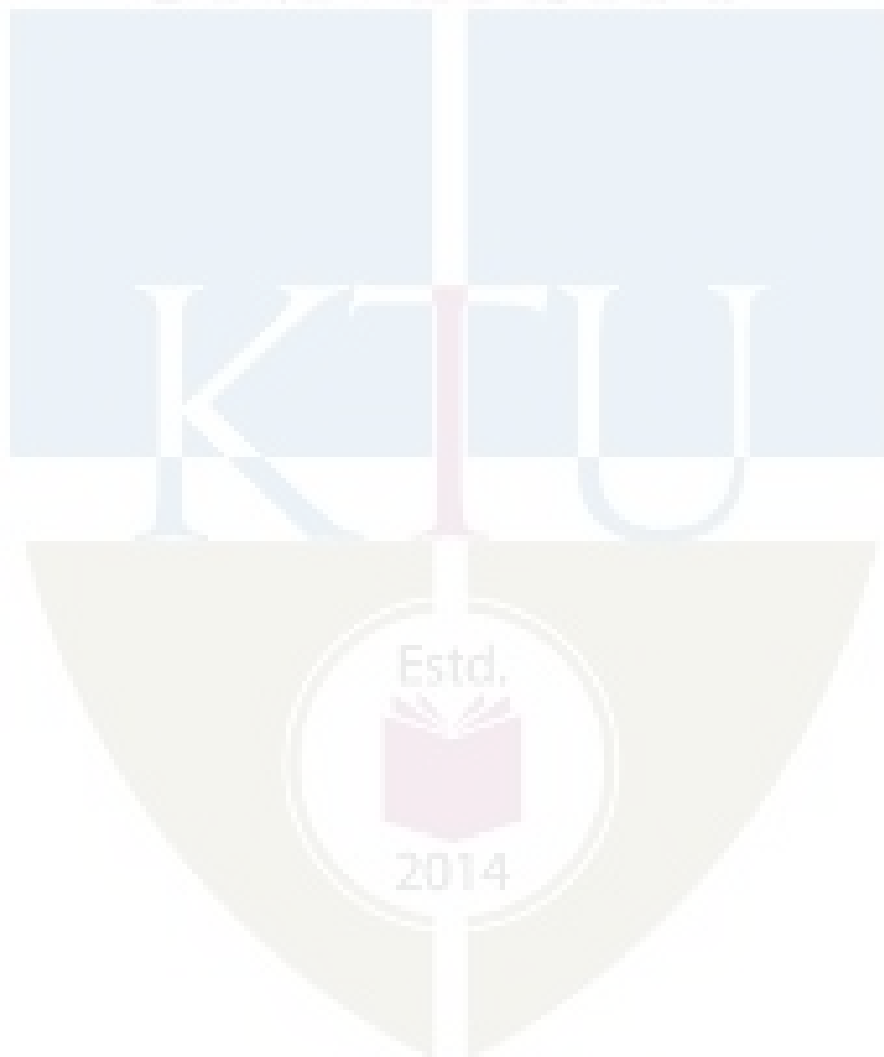
		cardinality of the hypothesis space)																																
19.	(a)	<p>Shown below is the Bayesian network corresponding to the Burglar Alarm problem, $P(J A) P(M A) P(A B, E) P(B) P(E)$. The probability tables show the probability that variable is True, e.g., $P(M)$ means $P(M = t)$. Find $P(J = t \wedge M = f \wedge A = f \wedge B = f \wedge E = t)$.</p> <pre> graph TD B((B)) --> A((A)) E((E)) --> A((A)) A((A)) --> J((J)) A((A)) --> M((M)) </pre> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>$P(E)$</td></tr><tr><td>.002</td></tr></table> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>A</td><td>$P(M)$</td></tr><tr><td>t</td><td>.70</td></tr><tr><td>f</td><td>.01</td></tr></table> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>B</td><td>E</td><td>$P(A)$</td></tr><tr><td>t</td><td>t</td><td>.95</td></tr><tr><td>t</td><td>f</td><td>.94</td></tr><tr><td>f</td><td>t</td><td>.29</td></tr><tr><td>f</td><td>f</td><td>.001</td></tr></table> <table border="1" style="display: inline-table;"> <tr><td>$P(B)$</td></tr><tr><td>.001</td></tr></table> <table border="1" style="display: inline-table;"> <tr><td>A</td><td>$P(J)$</td></tr><tr><td>t</td><td>.90</td></tr><tr><td>f</td><td>.05</td></tr></table>	$P(E)$.002	A	$P(M)$	t	.70	f	.01	B	E	$P(A)$	t	t	.95	t	f	.94	f	t	.29	f	f	.001	$P(B)$.001	A	$P(J)$	t	.90	f	.05	(7)
$P(E)$																																		
.002																																		
A	$P(M)$																																	
t	.70																																	
f	.01																																	
B	E	$P(A)$																																
t	t	.95																																
t	f	.94																																
f	t	.29																																
f	f	.001																																
$P(B)$																																		
.001																																		
A	$P(J)$																																	
t	.90																																	
f	.05																																	
	(b)	Derive an algorithm using the inverse transform method to generate a random sample from the distribution with density $f_X(x) = 3x^2, 0 < x < 1$.	(7)																															
OR																																		
20.	(a)	Draw the Bayesian Network that corresponds to this conditional probability: $P(A B,C,E) P(B D,E) P(C F,H) P(D G) P(E G,H) P(F H) P(G) P(H)$	(6)																															
	(b)	What is effective sample size (ESS)? Why is a large ESS necessary but not sufficient for good MCMC mixing?	(3)																															
	(c)	Describe the overall Gibbs sampling algorithm briefly	(5)																															

Teaching Plan

Module 1 : (Supervised Learning)(10 hours)		
1.1	Supervised, semi-supervised, unsupervised learning, reinforcement learning (TB 2: Ch 1)	1 hour
1.2	Least squares linear regression (TB 2: Section 2.6)	1 hour
1.3	Gradient descent, closed form, normal equations (TB 2: Section 5.8)	1 hour
1.4	Regularization techniques (LASSO, RIDGE) (TB 4: Section 7.1)	1 hour
1.5	Polynomial regression (TB 2: Section 2.6)	1 hour
1.6	Logistic Regression (TB 6: Section 3.3)	1 hour
1.7	Decision Tree Learning (ID3) (TB 8: Section 8.2)	1 hour
1.8	Decision Tree Learning (C4.5) (TB 8: Section 8.2)	1 hour
1.9	Naive Bayes Classifier (TB 8: Section 8.3)	1 hour
1.10	Gaussian Discriminant Analysis (GDA) (TB 7: Section 5.2,5.3)	1 hour
Module 2 : (Unsupervised Learning)(8 hours)		
2.1	Similarity measures (TB 8: Section 2.4)	1 hour
2.2	Hierarchical Agglomerative Clustering (TB 3: Chapter 14)	1 hour
2.3	Hierarchical Agglomerative Clustering (TB 3: Chapter 14)	
2.4	K-means partitional clustering (TB 3: Chapter 13)	1 hour
2.5	K-medoids partitional clustering	
2.6	Gaussian mixture models (TB 3: Chapter 13)	1 hour
2.7	Expectation Maximization (EM) algorithm for Gaussian mixture model Lecture-1 (TB 3: Chapter 13)	1 hour
2.8	Expectation Maximization (EM) algorithm for Gaussian mixture model Lecture-2 (TB 3: Chapter 13)	1 hour
Module 3 : (Practical aspects in machine learning) (6 hours)		

3.1	Precision, Recall, Accuracy, F-Measure, ROC, AUC (TB8.5/TB 3: Chapter 22.1)	1 hour
3.2	Generalisation and overfitting, cross-validation (TB 2: Section 2.7,4.8)	1 hour
3.3	Bias-variance tradeoff (TB 2: Chapter 22.3)	1 hour
3.4	Error estimation, parameter and model selection (TB 3: Chapter 8.5)	1 hour
3.5	Bagging, Boosting (TB 8: Chapter 8.6)	1 hour
3.6	Adaboost, Random Forests (TB 8: Chapter 8.6)	1 hour
Module 4 : (Statistical Learning Theory) (TB 5 – Chapter 2, 3.3)(7 hours)		
4.1	Learning in the limit, probably approximately correct (PAC) learning	1 hour
4.2	Quantifying the number of examples needed to PAC learn	1 hour
4.3	Computational complexity of training	1 hour
4.4	Sample complexity for finite hypothesis spaces	1 hour
4.5	PAC results for learning conjunctions	1 hour
4.6	Sample complexity for infinite hypothesis spaces	1 hour
4.7	Vapnik-Chervonenkis(VC) dimension	1 hour
Module 5 : (Advanced Machine Learning Topics) (13 hours)		
5.1	Bayesian belief networks (TB 1 – Chapter 8)	1 hour
5.2	Markov random fields (TB 1 – Chapter 8)	1 hour
5.3	Inference on chains and factor graphs (TB 1 – Chapter 8)	1 hour
5.4	Inference on clique trees (TB 1 – Chapter 8)	1 hour
5.5	Basic sampling algorithms (TB 1 – Chapter 11)	1 hour
5.6	Rejection sampling (TB 1 – Chapter 11)	1 hour
5.7	Importance sampling (TB 1 – Chapter 11)	1 hour
5.8	Markov chain Monte Carlo(MCMC) (TB 1 – Chapter 11)	1 hour
5.9	Gibbs sampling (TB 1 – Chapter 11)	1 hour

5.10	Variational method (TB 1 – Chapter 10)	1 hour
5.11	Auto Encoder (TB 4 – Chapter 14)	1 hour
5.12	Variational AutoEncoder (TB 9 – Chapter 3)	1 hour
5.13	Generative Adversarial Networks (TB 9 – Chapter 4)	1 hour



CST 398	THEORY OF COMPUTABILITY AND COMPLEXITY	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This is a theoretical course in computer science to enable the learners to know the fundamentals of computability and complexity theories. It covers the notions of computability/decidability, the process of reduction to prove decidability/undecidability and the classification of problems into class P, class NP and class NP Complete based on the time complexity of solving the problems. This course helps the learner to identify whether a real life problem is decidable/undecidable and also to classify a decidable problem into tractable or intractable, based on the time complexity class it belongs.

Prerequisite: Sound knowledge in Data Structures and Formal Languages and Automata Theory.

Mapping of course outcomes with program outcomes

CO1	Illustrate relative computing powers of Finite State Automata, Push Down Automata, Linear Bounded Automata and Turing Machines.(Cognitive Knowledge Level: Apply)
CO2	Prove that a given language is undecidable/not semi-decidable by using the reduction process.(Cognitive Knowledge Level: Apply)
CO3	Describe the time complexity of a given problem as a function of the number of steps required by a Turing machine to solve it. (Cognitive Knowledge Level: Understand)
CO4	Utilize polynomial time reduction to prove that a given problem is NP Complete. (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO 1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												

CO3												
CO4												

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	10 marks
Continuous Assessment Tests(Average of Internal Tests1&2)	25 marks
Continuous Assessment Assignment	15 marks

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus**Module - 1 (Introduction to Formal Language Theory and Regular Languages)**

Finite State Automata, Push Down Automata, Linear Bounded Automata, Turing Machines, Recursive Languages, Recursively Enumerable Languages, Universal Turing Machine, Enumeration Machine, Two Counter Machine.

Module- 2 (Undecidability)

Halting Problem, Language representation of a problem, Reduction - applications, Rice's First and Second Theorem with proof.

Module - 3 (Overview of Complexity Classes)

Measuring time complexity, Asymptotic notations - Big O and small-o, Analysing algorithms, Complexity relationship among models. Complexity classes- Class P, example problems in class P, Class NP, Polynomial time verification, example problems in class NP.

Module- 4 (NP Completeness)

Satisfiability problem, Polynomial time reducibility, Overview of Graphs, NP Complete Problems, Cook-Levin theorem (SAT is NP Complete).

Module- 5 (More NP Complete Problems)

CLIQUE, Vertex Cover and Hamiltonian path with proof of correctness of NP Completeness.

Text Books

1. Dexter C. Kozen, Automata and Computability, Springer (1999)
2. Michael Sipser, Introduction to the Theory of Computation, Second Edition

Reference Books

1. Douglas B. West, Introduction to Graph Theory, Second Edition

Course Level Assessment Questions**Course Outcome1 (CO1):**

Identify the class of the following languages in Chomsky Hierarchy:

1. Design a Finite State Automaton for the language $L = \{axb|x \in \{a, b\}^*\}$
2. Design a Push Down Automaton for the language $L = \{a^n b^n | n \geq 0\}$
3. Design a Linear Bounded Automaton for the language $L = \{a^n b^n c^n | n \geq 0\}$
4. Design a Turing Machine for the language $L = \{ww|w \in \{a, b\}^*\}$

Course Outcome 2(CO2):

Without using Rice's Theorem prove that neither the set FIN (refer Text Book 1) nor its complement is recursively enumerable.

Course Outcome 3(CO3):

Show that the language $L = \{a^n b^n | n \geq 0\}$ can be decided by a deterministic Turing Machine in quadratic time.

Course Outcome 4(CO4): .

Using polynomial time reduction, prove that SUBSET-SUM (refer Text Book 2) problem is NP Complete.

Model Question Paper

QP CODE:

Reg No: _____

Name: _____

PAGES : 4

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH. DEGREE EXAMINATION(HONORS), MONTH & YEAR

Course Code: CST 398

Course Name: Theory of Computability and Complexity

Max.Marks:100

Duration: 3 Hours

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Design a Deterministic Finite state Automaton (DFA) for the language: $L = \{x \in \{0,1\}^* | x \text{ does not contain consecutive zeros}\}$.
2. Design a Pushdown Automaton (PDA) for the language $L = \{a^m b^n | m \geq 0 \text{ and } n > m\}$ (no explanation is required, just list the transitions in the PDA).
3. List any *six* undecidable problems.
4. Illustrate how a problem can be represented as a language.
5. Describe the term time complexity class.
6. Define the term polynomial time verification. Describe its usage.
7. Define the term polynomial time reduction. Describe its usage.
8. Define vertex cover. Illustrate with the help of a graph.

9. Illustrate CLIQUE problem with an example.
10. State Hamiltonian path problem. Show an example.

(10x3=30)

Part B**(Answer any one question from each module. Each question carries 14 Marks)**

11. (a) Why the family of languages recognized by Turing machines is called Recursively Enumerable? Explain the working of an enumeration machine. (8)
- (b) Illustrate the functioning of a Universal Turing Machine. (6)
- OR**
12. (a) Illustrate the functioning of a two counter machine. (4)
- (b) Prove that Turing Machines and Two Counter Machines are equivalent in power. (10)
13. (a) Prove using Diagonalisation that halting problem is undecidable. (8)
- (b) Prove Using reduction that state entry problem of Turing machines is undecidable. (6)
- OR**
14. (a) State and prove Rice's first theorem. (8)
- (b) Prove Using reduction that whether a Turing Machine accepts empty string (or null string) is undecidable. (6)
15. (a) Show that the language $L = \{a^n b^n | n \geq 0\}$ can be decided by a deterministic Turing Machine in $O(n \cdot \log n)$ time. (7)
- (b) Let $t(n)$ be a function, where $n \in \mathbb{N}$ and $t(n) \geq n$. Then, prove that every $t(n)$ time nondeterministic single-tape Turing machine has an equivalent $2^{O(t(n))}$ time single-tape deterministic Turing machine. (7)

OR

16. (a) Prove that every context free language is a member of class P. (8)
- (b) When is a problem said to be in class NP? (6)
Prove that Hamiltonian path problem of a directed graph is in class NP.
17. (a) Define Independent set in a graph. Prove that a graph G of n vertices with an independent set of size k contains a vertex cover of size $n - k$. (8)
- (b) Define the complexity class NP Complete. Explain the significance of an NP Complete problem. (6)

OR

18. (a) Define the complement of a graph. Prove that the complement of a graph G of n vertices with a CLIQUE of size k contains an independent set of size k . (7)
- (b) What is satisfiability problem. Prove that satisfiability problem is in class NP. (7)
19. (a) Illustrate Hamiltonian path in a Graph. (4)
- (b) Prove that Hamiltonian path problem is in the class NP Complete. (10)

OR

20. (a) Prove that Vertex Cover problem is in the class NP Complete. (8)
- (b) Why is it useful to identify that a problem is in the class NP Complete? (6)

Teaching Plan

No	Contents	No. of Lecture Hours (45 hrs)
Module-1(Overview of Automata Theory) (10 hours)		
1.1	Finite State Automata	1 hour
1.2	Push Down Automata	1 hour
1.3	Linear Bounded Automata	1 hour
1.4	Turing Machines	1 hour
1.5	Recursive Languages	1 hour
1.6	Recursively Enumerable Languages	1 hour
1.7	Universal Turing Machine	1 hour
1.8	Enumeration Machine	1 hour
1.9	Two Counter Machines	1 hour
1.10	Proof that two Counter Machines and Turing machines are equivalent	1 hour
Module-2 (Undecidability) (10 hours)		
2.1	Halting problem of Turing machine	1 hour
2.2	Proof of undecidability of Halting Problem	1 hour
2.3	Language representation of a problem	1 hour
2.4	Reduction	1 hour
2.5	Applications of reduction - Lecture 1	1 hour

2.6	Applications of reduction - Lecture 2	1 hour
2.7	Rice's First Theorem	1 hour
2.8	Proof of Rice's First Theorem	1 hour
2.9	Rice's Second Theorem	1 hour
2.10	Proof of Rice's Second Theorem	1 hour
Module-3 (Overview of Complexity Classes) (10 hours)		
3.1	Measuring time complexity, Asymptotic notations - Big O and small-o	1 hour
3.2	Analysing algorithms - time complexity class	1 hour
3.3	Complexity relationship among models - Single tape Turing Machine	1 hour
3.4	Multi-tape Turing Machine, Nondeterministic Turing Machine	1 hour
3.5	Class P	1 hour
3.6	Example problems in Class P	1 hour
3.7	Class NP	1 hour
3.8	Polynomial time verification	1 hour
3.9	Example problems in Class NP - Lecture 1	1 hour
3.10	Example problems in Class NP - Lecture 2	1 hour
Module-4 (NP Completeness) (9 hours)		
4.1	Satisfiability problem	1 hour
4.2	Polynomial time reducibility	1 hour
4.3	Overview of Graphs, CLIQUE, INDEPENDENT SET	1 hour
4.4	Vertex Cover	1 hour
4.5	Reducing 3SAT problem to CLIQUE - Lecture 1	1 hour
4.6	Reducing 3SAT problem to CLIQUE - Lecture 2	1 hour
4.7	NP Complete Problems	1 hour
4.8	Cook-Levin theorem, Proof - Lecture 1	1 hour
4.9	Proof - Lecture 2	1 hour
Module-5 (More NP Complete Problems) (6 hours)		

5.1	CLIQUE	1 hour
5.2	Vertex Cover - Lecture 1	1 hour
5.3	Vertex Cover - Lecture 2	1 hour
5.4	Hamiltonian path - Lecture 1	1 hour
5.5	Hamiltonian path - Lecture 2	1 hour
5.6	Hamiltonian path - Lecture 3	1 hour

