Data Structures – CST 201
Module - 3

Syllabus

- **Linked List and Memory Management**
  - Self Referential Structures
  - Dynamic Memory Allocation
  - Singly Linked List-Operations on Linked List.
  - Doubly Linked List
  - Circular Linked List
  - **Stacks using Linked List**
  - Queues using Linked List
  - Polynomial representation using Linked List
  - Memory allocation and de-allocation
    - First-fit, Best-fit and Worst-fit allocation schemes

Stack Data Structure

- **Definition:** A stack is an ordered collection of homogeneous data elements where the insertion and deletion operations take place only at one end called top of the stack
- Stack is a **Last-in-First-Out(LIFO)** data structure. Items are removed in the reverse order from that in which they were inserted. It is also called LIFO list.
- **Basic operations on stack:**
  - PUSH: Insert an item at the top of stack
  - POP: Delete an item from the top of stack
- **Stack Representations:**
  - Array Representation
  - Linked List Representation

Stack Using Linked List

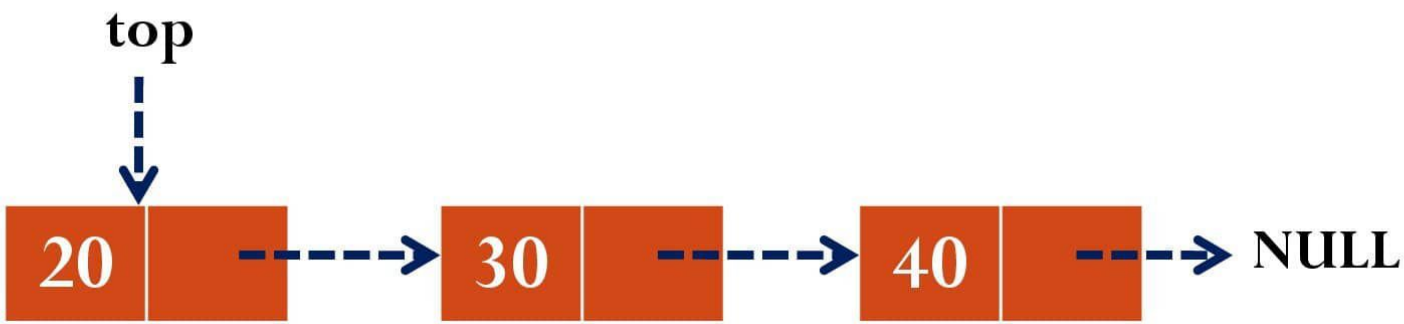
- Data structure used is **Singly Linked List**

- **PUSH**: Insert an item in the beginning of the list
- **POP**: Delete an item from the beginning of the list

PUSH - Algorithm

Algorithm PUSH(top, item)

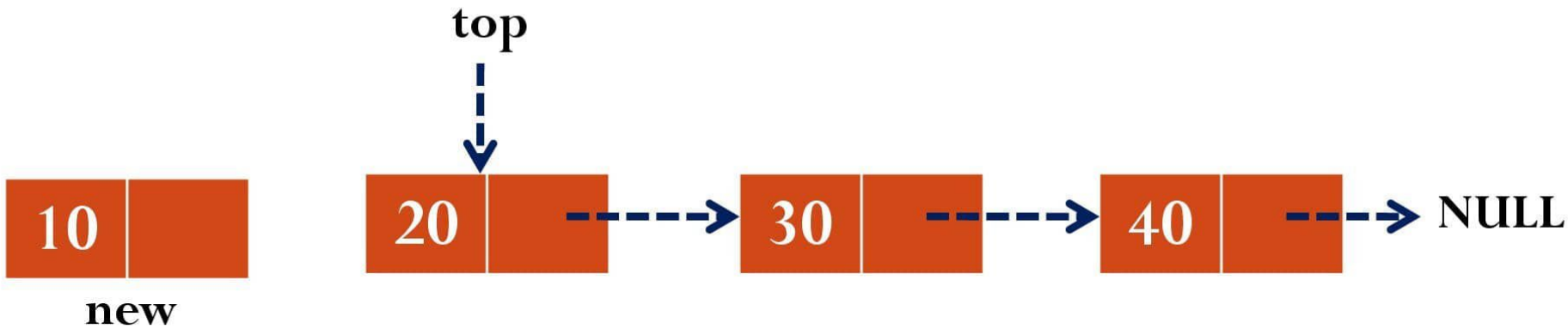
1. Create a node new
2. new→data=item
3. new→link=top
4. top=new



PUSH - Algorithm

Algorithm PUSH(top, item)

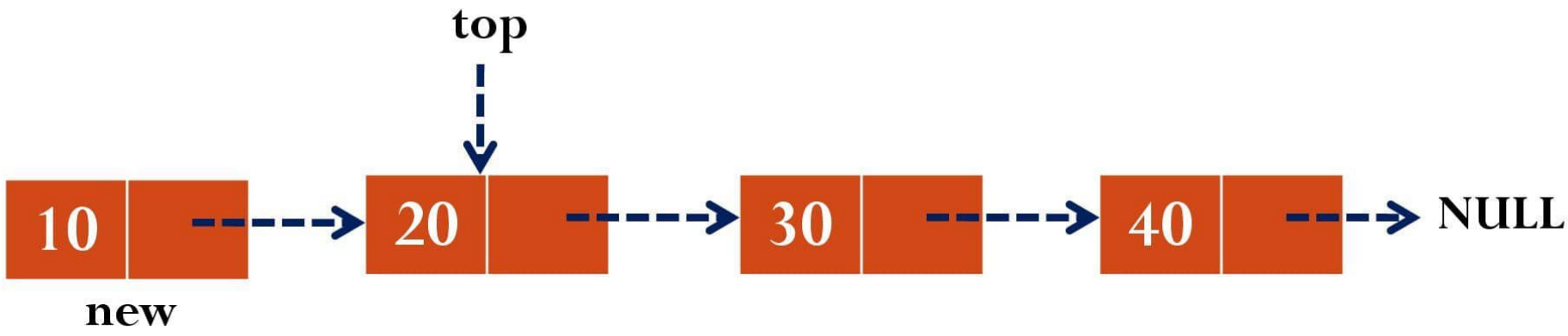
1. Create a node new
2. new→data=item
3. new→link=top
4. top=new



PUSH - Algorithm

Algorithm PUSH(top, item)

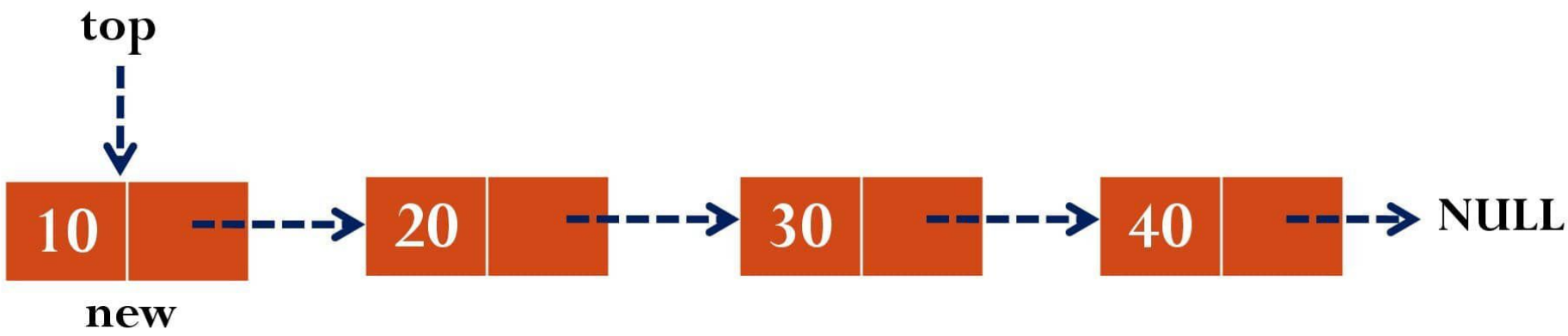
1. Create a node new
2. new→data=item
3. new→link=top
4. top=new



PUSH - Algorithm

Algorithm PUSH(top, item)

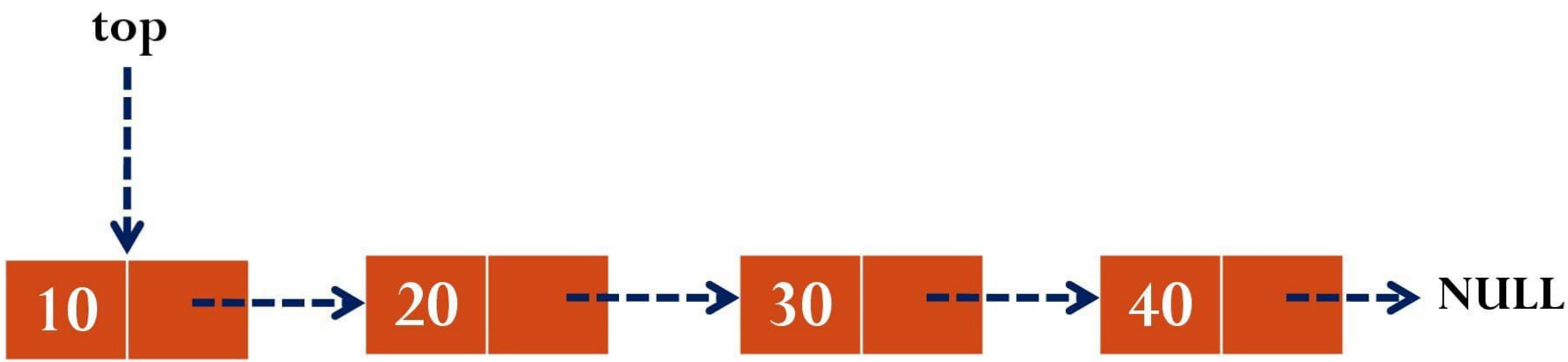
1. Create a node new
2. new→data=item
3. new→link=top
4. top=new



POP - Algorithm

Algorithm POP(top)

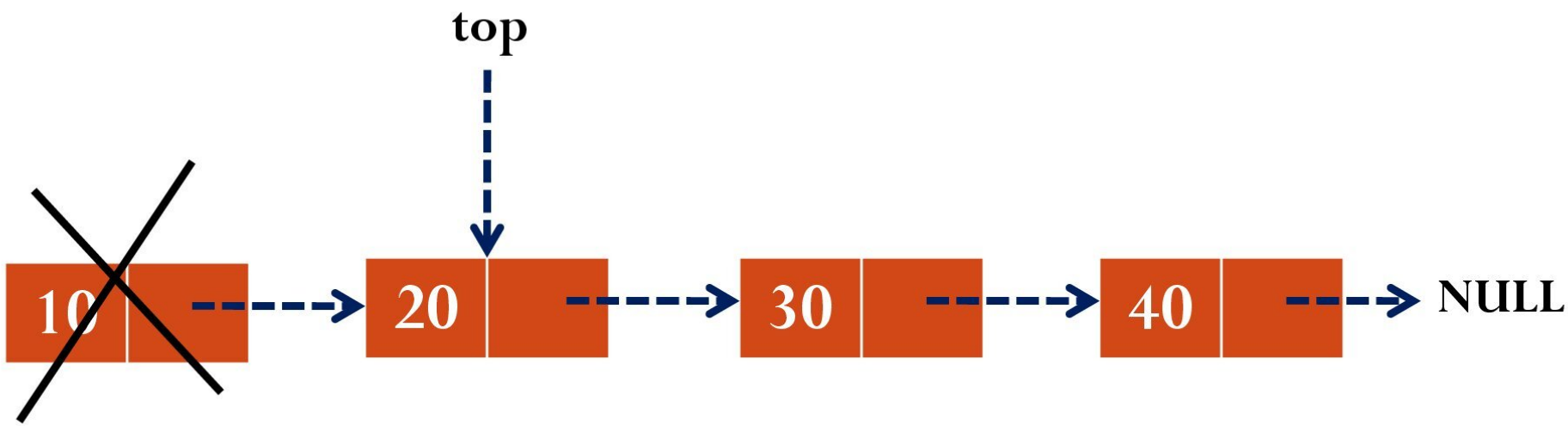
1. If top=NULL then
    1. Print "Stack is Empty"
2. Else
    1. Print "Poped item is " top→data
    2. top=top→link



POP - Algorithm

Algorithm POP(top)

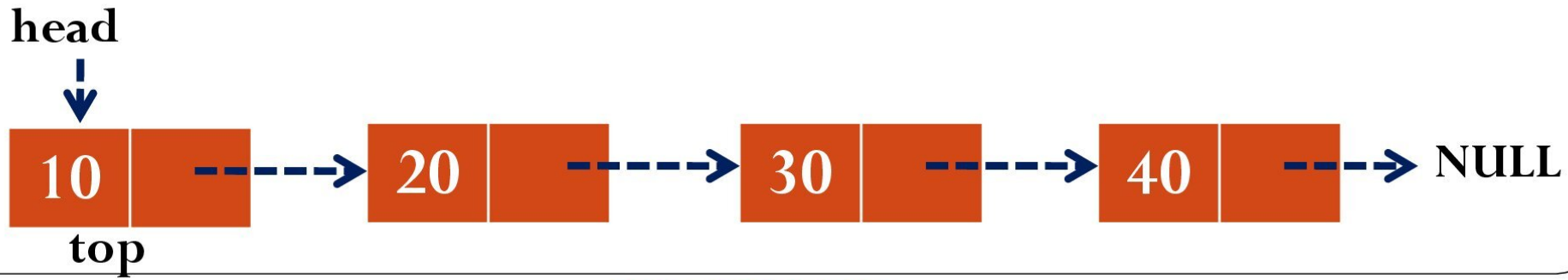
1. If top=NULL then
    1. Print "Stack is Empty"
2. Else
    1. Print "Poped item is " top→data
    2. top=top→link



Display - Algorithm

Algorithm Display(top)

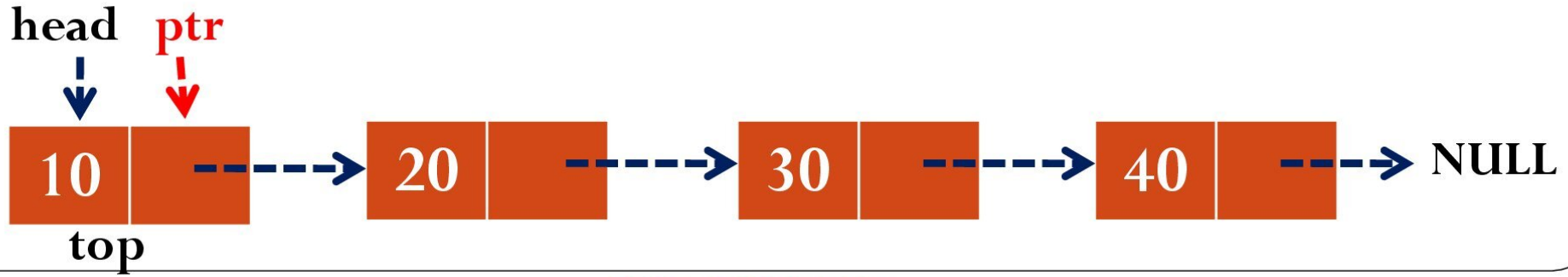
1. If top=NULL then
    1. Print “Stack is Empty”
2. Else
    1. ptr=top
    2. While ptr!=NULL do
        1. Print ptr→data
        2. ptr=ptr→link



Display - Algorithm

Algorithm Display(top)

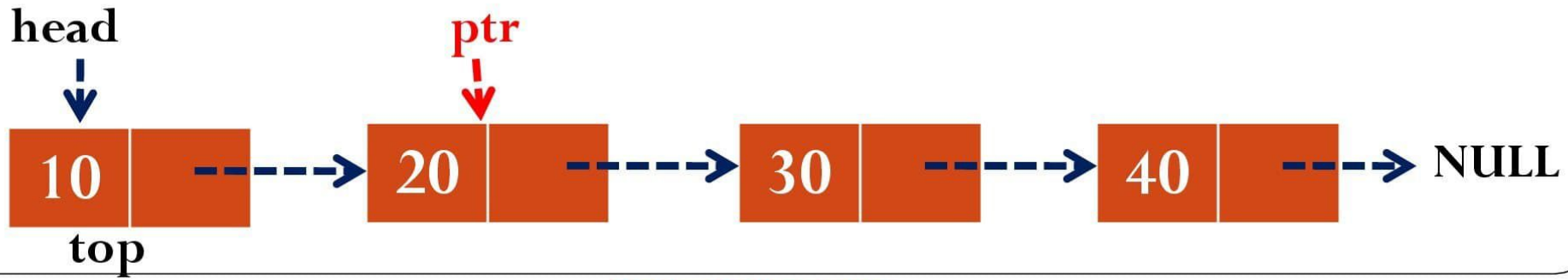
1. If top=NULL then
    1. Print “Stack is Empty”
2. Else
    1. ptr=top
    2. While ptr!=NULL do
        1. Print ptr→data
        2. ptr=ptr→link



Display - Algorithm

Algorithm Display(top)

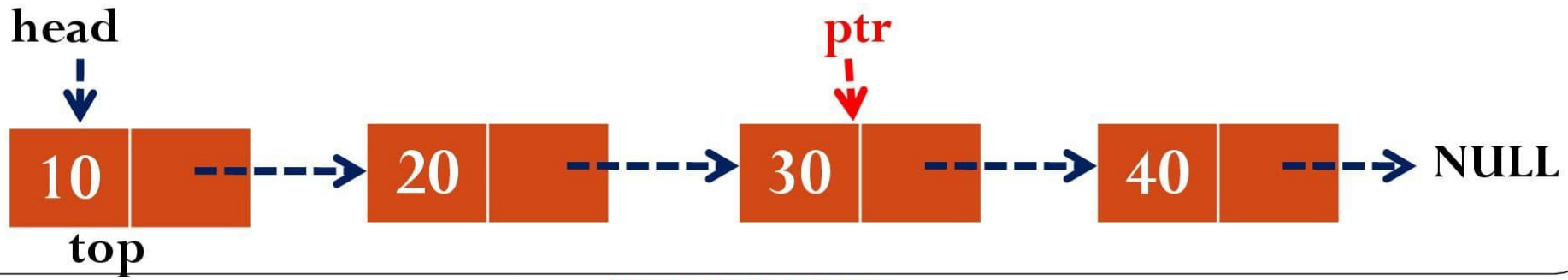
1. If top=NULL then
    1. Print "Stack is Empty"
2. Else
    1. ptr=top
    2. While ptr!=NULL do
        1. Print ptr→data
        2. ptr=ptr→link



Display - Algorithm

Algorithm Display(top)

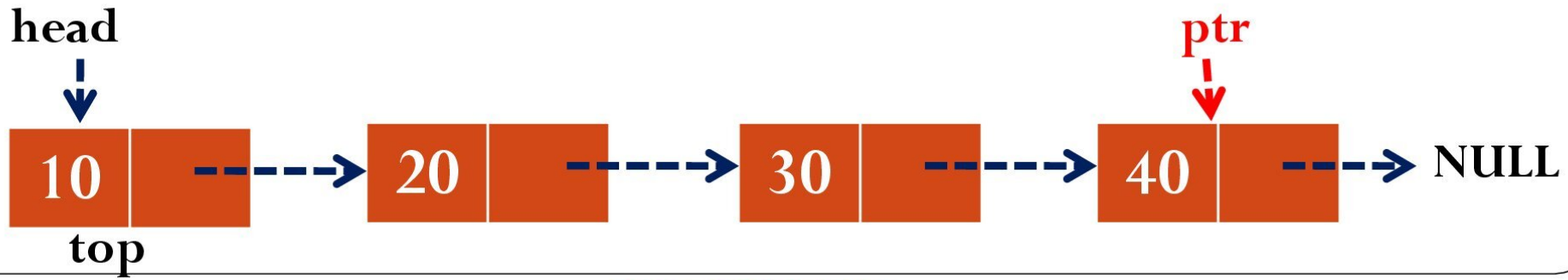
1. If top=NULL then
    1. Print "Stack is Empty"
2. Else
    1. ptr=top
    2. While ptr!=NULL do
        1. Print ptr→data
        2. ptr=ptr→link



Display - Algorithm

Algorithm Display(top)

1. If top=NULL then
    1. Print "Stack is Empty"
2. Else
    1. ptr=top
    2. While ptr!=NULL do
        1. Print ptr→data
        2. ptr=ptr→link



Display - Algorithm

Algorithm Display(top)

1. If top=NULL then
    1. Print "Stack is Empty"
2. Else
    1. ptr=top
    2. While ptr!=NULL do
        1. Print ptr→data
        2. ptr=ptr→link