

Data Structures and Algorithms

Prepared by Prof. Shereen Taie

IT251
Information Technology Department

Lecture 5



Data Structures and Algorithms

Lecture 5

IT251

Information Technology Department

Prepared by
Prof. Shereen Taie

www.midocean.ae

DATA STRUCTURES

Chapter

03

Data Structures:

- Data may be organized in many different ways logical or mathematical model of a program particularly organization of data. This organized data is called “Data Structure”.

Or

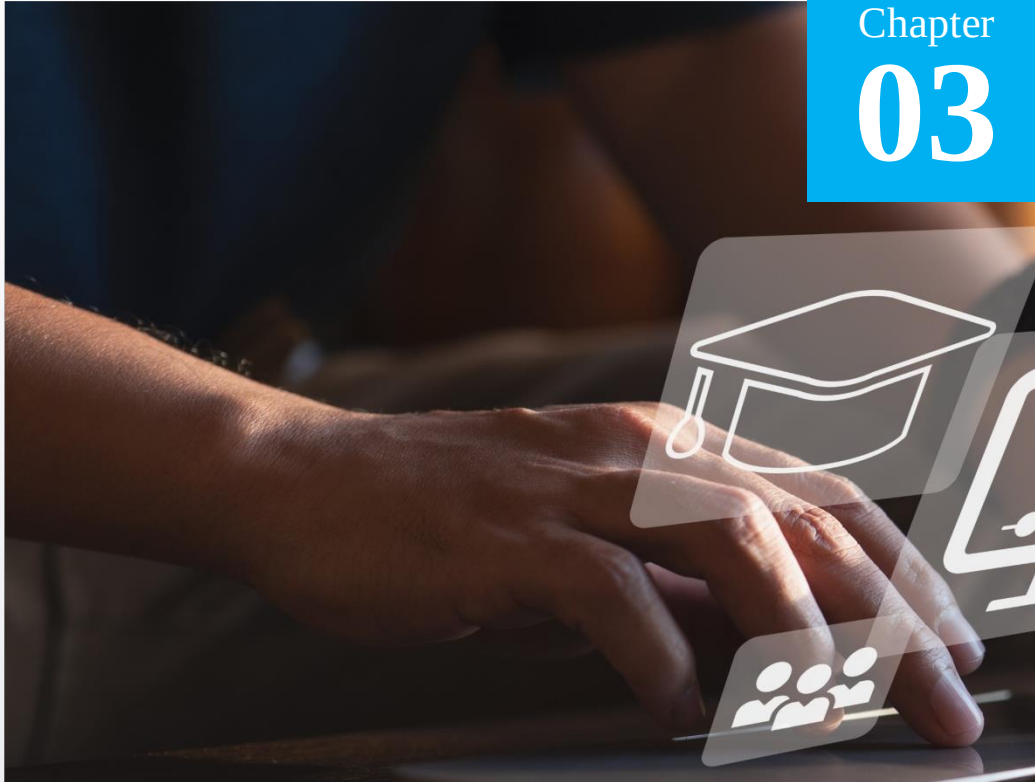
- The organized collection of data is called a ‘Data Structure’.

Data Structure = Organized data + Allowed operations

Chapter

03

Data Structure Goals



Data Structure Goals:

- **Data Structure involves two complementary goals:**

The first goal:

is to identify and develop useful, **mathematical entities** and **operations**

and to determine **what class of problems can be solved** by using these

entities and operations.

Data Structure Goals:

- **Data Structure involves two complementary goals:**

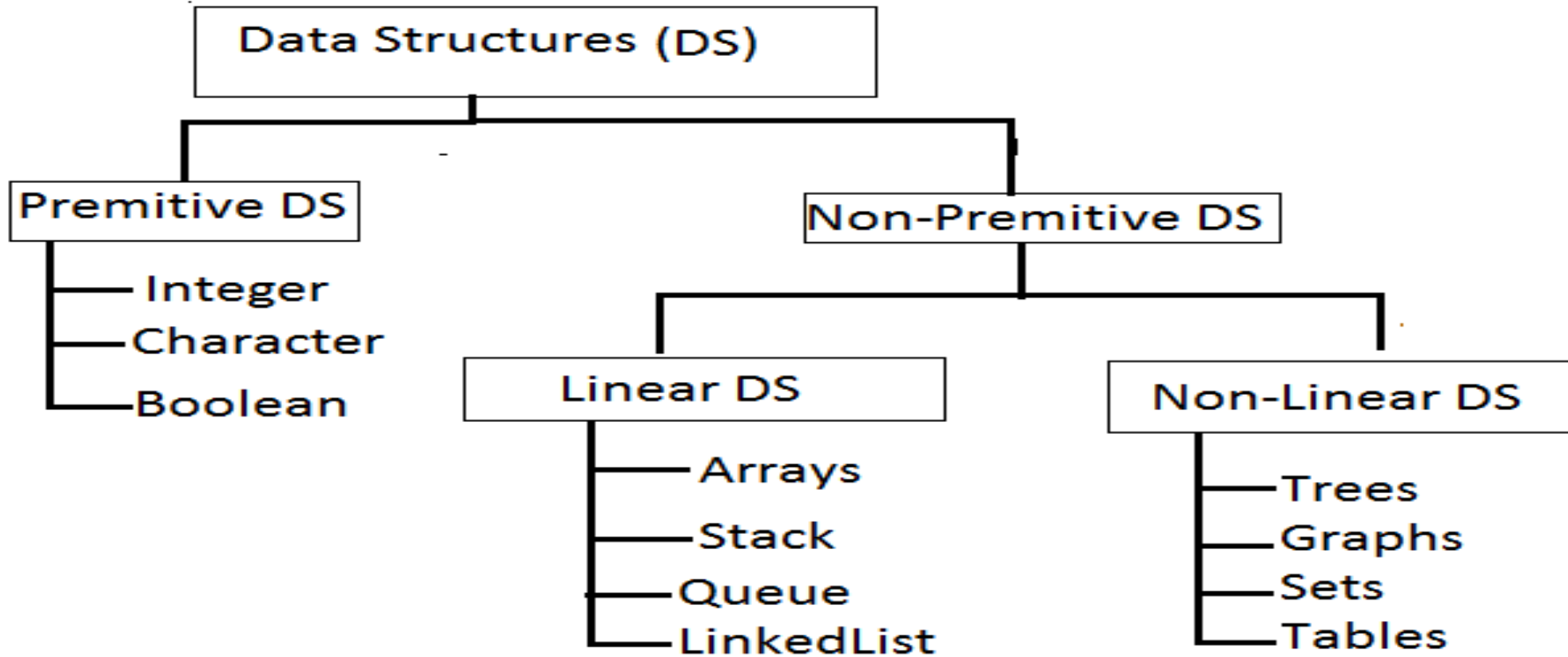
The second goal:

is to **determine representation for those abstract entities** to implement
abstract operations on this concrete representation.

Data Structure Categories

Chapter

03



Important for all other branches of computer science

Primitive Data structures:

- **Primitive Data structures** are directly supported by the programming language.
- i.e; any operation is directly performed in these data items.
- Ex: integer, Character, Real numbers etc.

Non-primitive data structures:

- **Non-primitive data types** are not defined by the programming language, but are instead created by the programmer.

Non-primitive data structures subcategories:

1. Linear data structures.
2. Non-Linear data structures.

Linear Data structures:

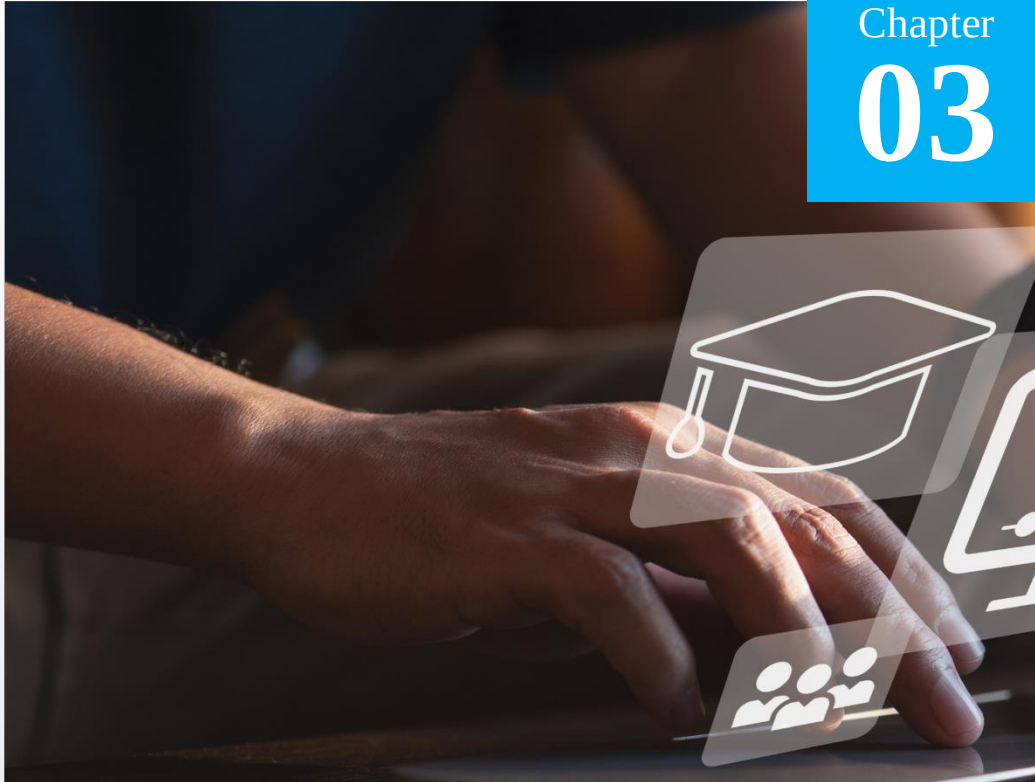
- Linear data structures **organize their data elements in a linear fashion**, where data elements are attached one after the other.
- Linear data structures are **very easy to implement**, since **the memory of the computer is also organized in a linear fashion**.
- Some commonly used linear data structures are arrays, linked lists, stacks and queues.

Nonlinear Data structures:

- In nonlinear data structures, data elements are not organized in a sequential fashion.
- Data structures like multidimensional arrays, trees, graphs, tables and sets are some examples of widely used nonlinear data structures.

Chapter
03

Operations on the Data Structures



Operations on the Data Structures :

Following operations can be performed on the data structures:

1. Traversing
2. Searching
3. Inserting
4. Deleting
5. Sorting
6. Merging

Operations on the Data Structures :

1. **Traversing** عبور: It is used to access each data item exactly once so that it can be processed.
2. **Searching**: It is used to find out the location of the data item if it exists in the given collection of data items.

Operations on the Data Structures :

3. **Inserting:** It is used to add a new data item in the given collection of data items. queues.
4. **Deleting:** It is used to delete an existing data item from the given collection of data items.

Operations on the Data Structures :

5. **Sorting:** It is used to arrange the data items in some order i.e. **in ascending or descending order** in case of numerical data and in **dictionary order** in case of alphanumeric data.

6. **Merging:** It is used to combine the data items of two sorted files into single file in the sorted form.

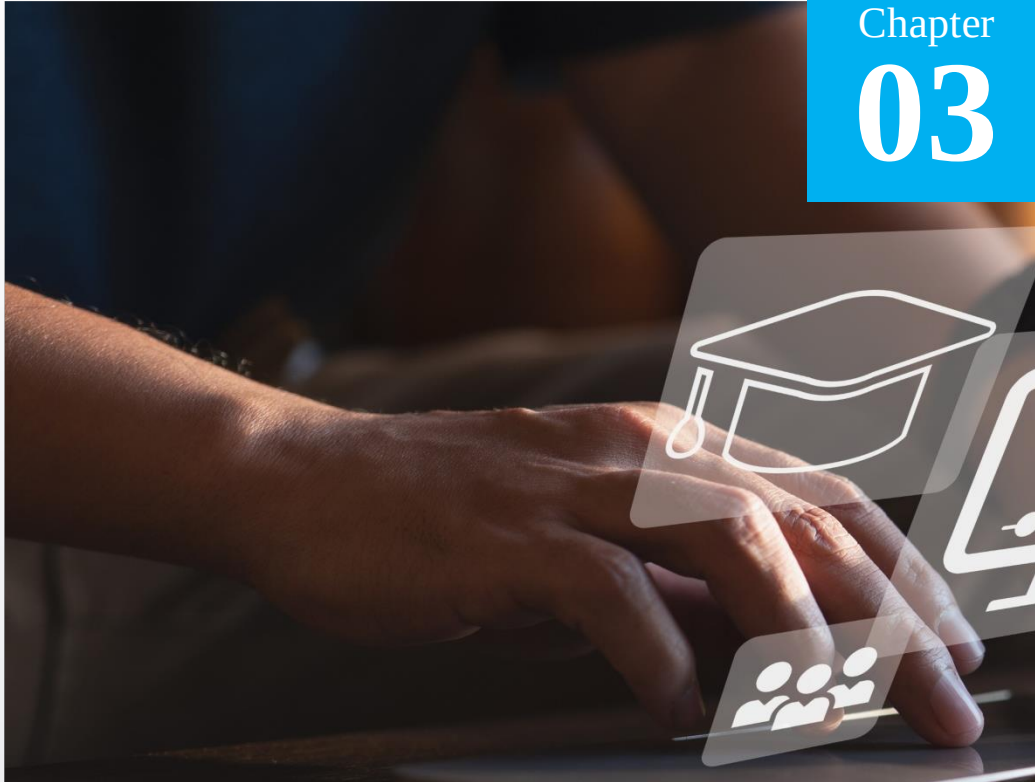
STACKS AND QUEUES

Chapter

03

Chapter
03

STACKS AND QUEUES



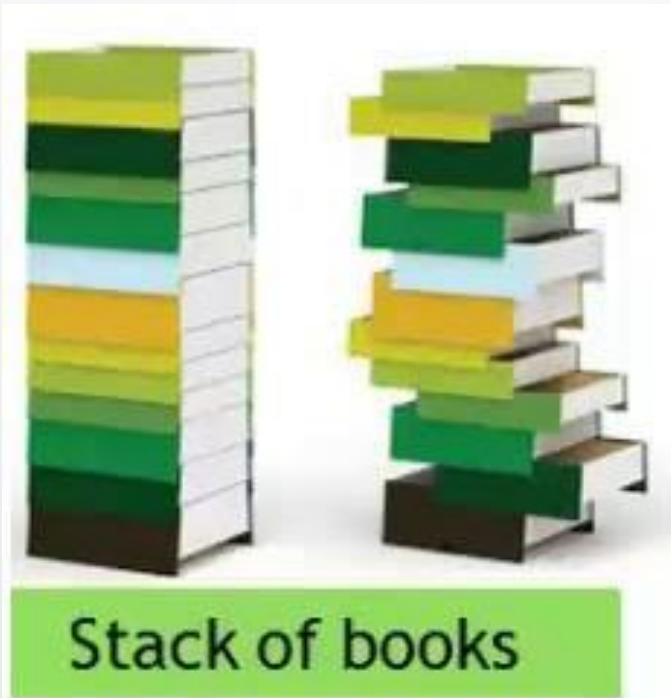
STACKS:

- A Stack is linear data structure.
- A stack is a list of elements in which an element may be inserted or deleted only at one end, called the top of the stack.

STACKS:

- Stack principle is LIFO (last in, first out). Which **element inserted last** on to the stack **that element deleted first from the stack**.
- As the items can be added or removed only from the top i.e. the last item to be added to a stack is the first item to be removed.

Real life examples of stacks are:



Operations on stack:

- The two basic operations associated with stacks are:
 1. Push
 2. Pop

Operations on stack:

- While performing push and pop operations the following test must be conducted on the stack.
 - a. Stack is empty or not
 - b. Stack is full or not

1. Push:

Push:

Push operation is used to add new elements in to the stack. At the time of addition first check the stack is full or not. If the stack is full it generates an error message "stack overflow".

2. Pop:

Pop:

Pop operation is used to delete elements from the stack. At the time of deletion first check the stack is empty or not. If the stack is empty it generates an error message "stack underflow".

Representation of Stack (or) Implementation of stack

The stack should be represented in two ways:

1. Stack using array
2. Stack using linked list

1. Stack using array:

- Let us consider a stack with 6 elements capacity.
- This is called as the size of the stack.
- The number of elements to be added should not exceed the maximum size of the stack.

1. Stack using array:

- Let us consider a stack with 6 elements capacity.
- This is called as the size of the stack.
- The number of elements to be added should not exceed the maximum size of the stack.

Stack Overflow :

- Similarly, you cannot remove elements beyond the base of the stack.
- If such is the case, we will reach a ***stack underflow condition***.

Operation Performed in Stack

push & pop & display

1.push():

push():

When an element is added to a stack, the operation is performed by

push().

Below Figure shows the creation of a stack and addition of elements using push()

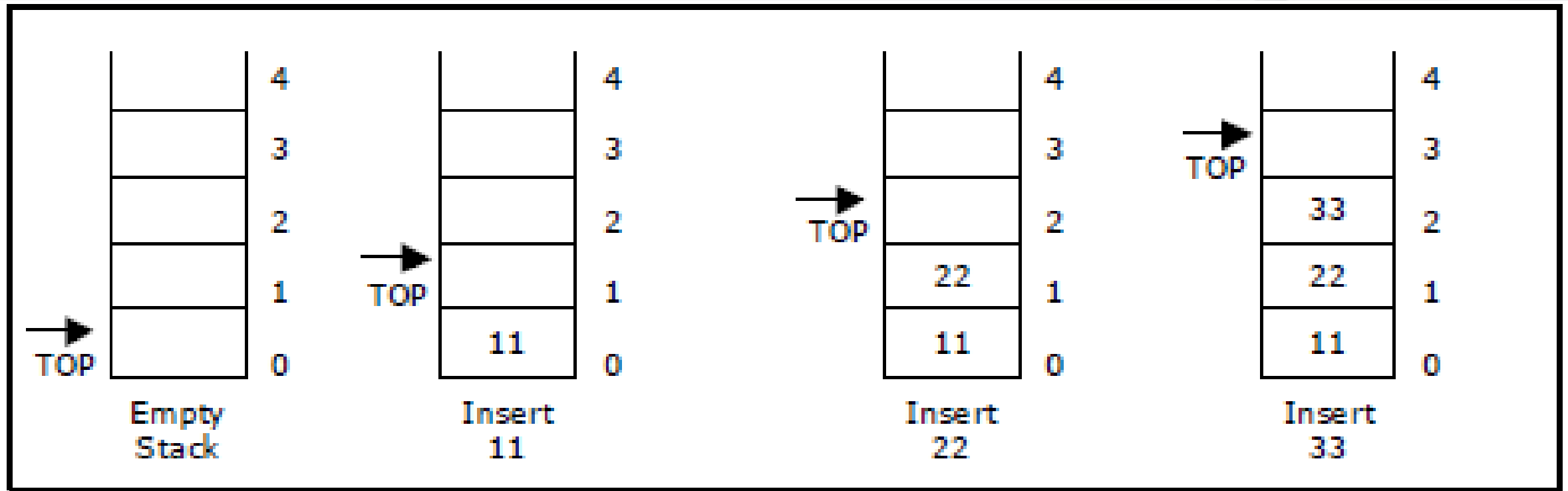


Figure . Push operations on stack

2.pop():

pop():

When an element is taken off from the stack, the operation is performed by

pop().

Below figure shows a stack initially with three elements and shows the deletion of elements using pop()

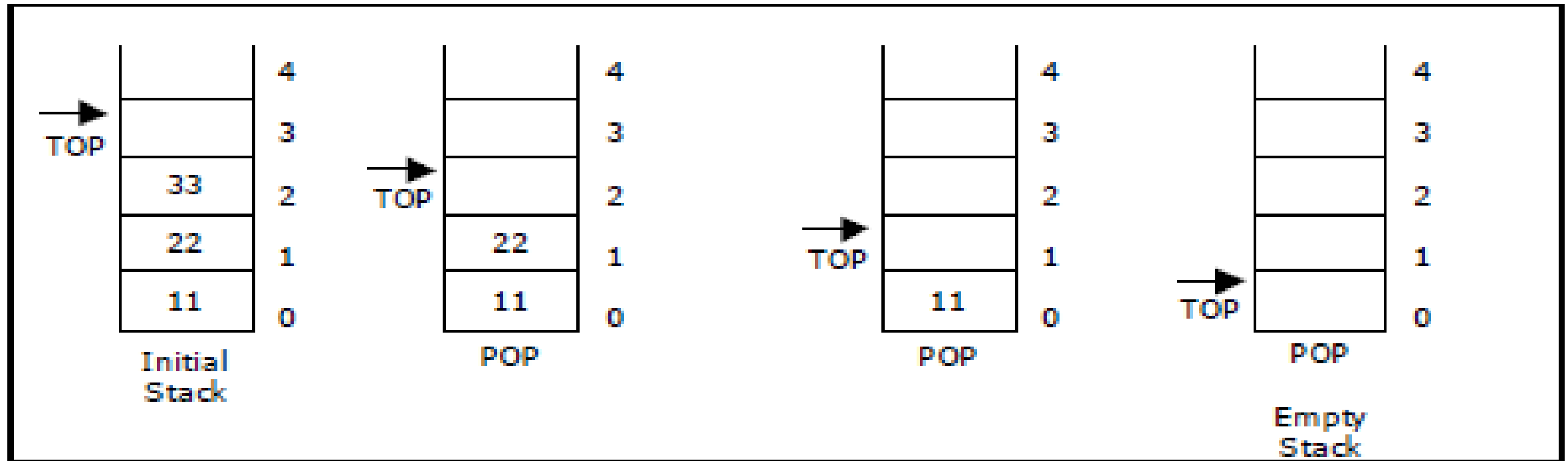


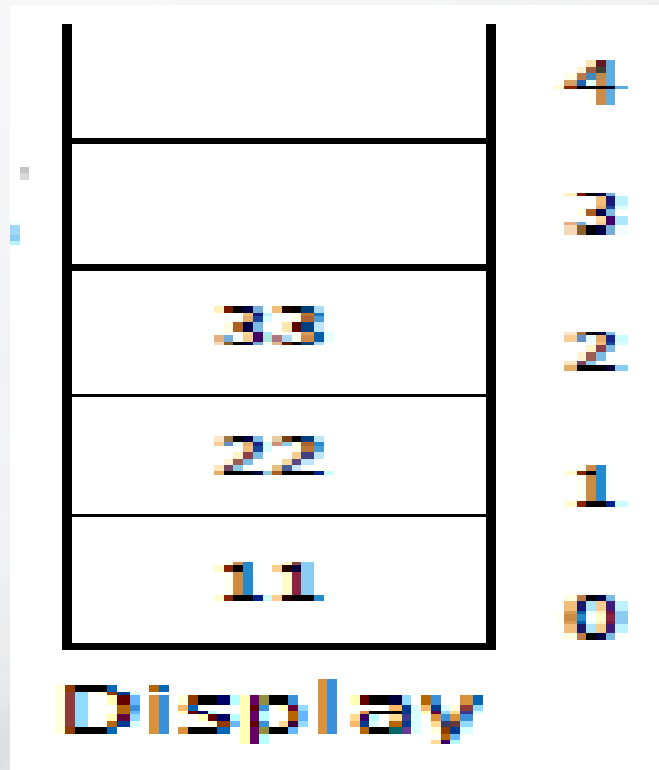
Figure Pop operations on stack

3. display():

display():

- This operation performed display the elements in the stack.
- We display the element in the stack check the condition is stack is empty or not, otherwise display the list of elements in the stack.

Display the elements in the stack





Thank You