Data Structure & Algorithm

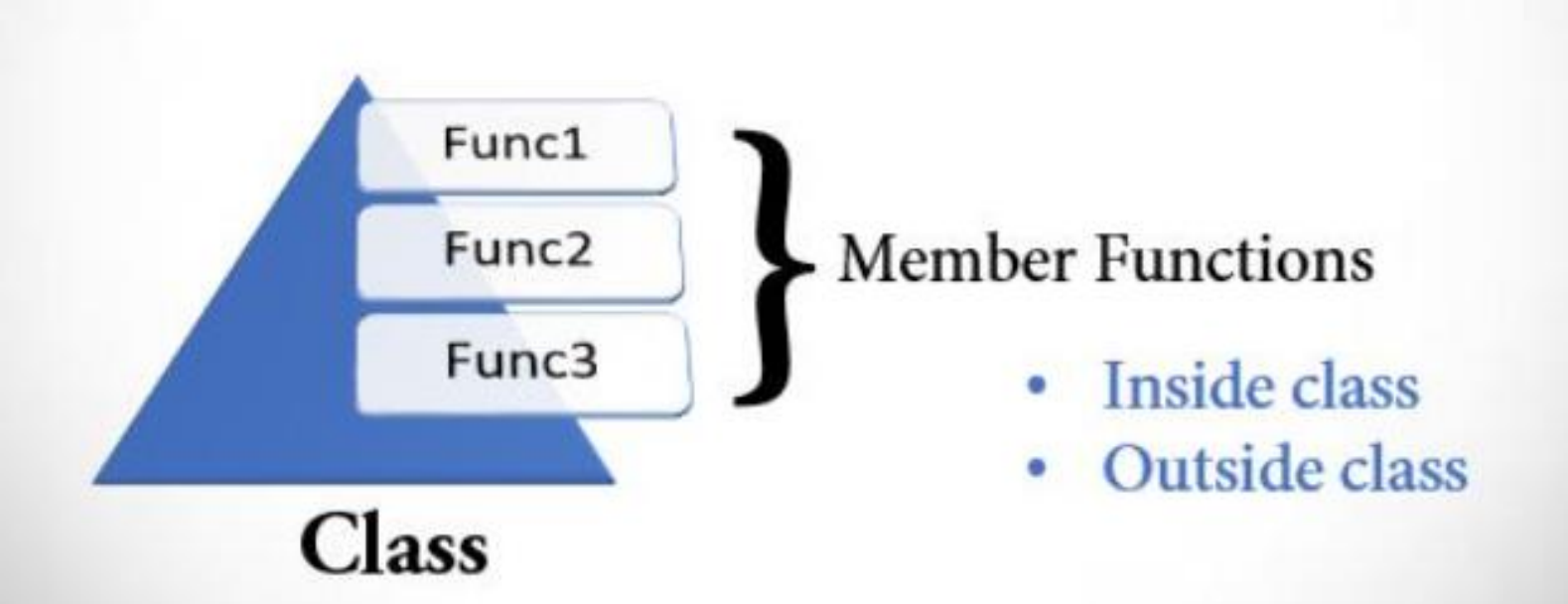
Lecture 6
Abstract Data Types: Stacks

Chhoeum Vantha, Ph.D.
Telecom & Electronic Engineering

Content

- Class and Functions in C++
- Abstract Data Types
  - Stacks
  - Queues and Priority Queues
  - Linked Lists
  - Abstract Data Types
  - Specialized Lists

Class and Functions in C++



Class: Member Function

- Inside the class definition

keyword user-defined name

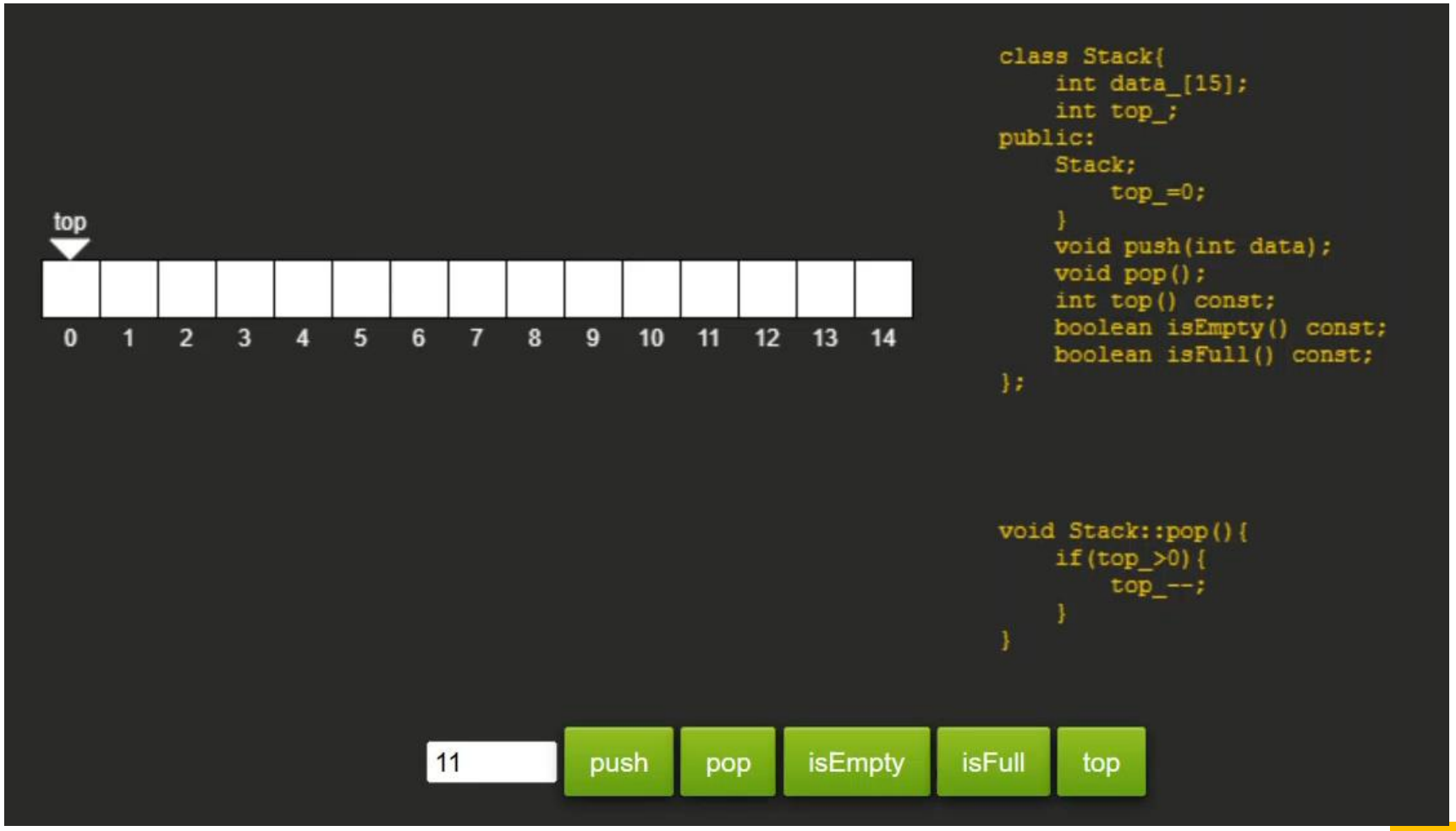
```
class ClassName

{  Access specifier:          //can be private,public or protected

   Data members;              // Variables to be used

   Member Functions() { }     //Methods to access data members

};                            // Class name ends with a semicolon
```

Class: Member Function

- Outside the class definition

```
class class_name
 {
   ........
   ........
   public:
     return_type function_name (args); //function declaration
 };
//function definition outside class
return_type class_name :: function_name (args)
 {
   ...........; // function definition
 }
```

What is the video all about?



Abstract Data Types?

- Abstract Data type (ADT) is a type (or class) for objects whose behavior is defined by a set of values and a set of operations.
- The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented.

The List ADT Functions is given below:

- get() – Return an element from the list at any given position.
- insert() – Insert an element at any position of the list.
- remove() – Remove the first occurrence of any element from a non-empty list.
- removeAt() – Remove the element at a specified location from a non-empty list.

The List ADT Functions is given below:

- replace() – Replace an element at any position by another element.
- size() – Return the number of elements in the list.
- isEmpty() – Return true if the list is empty, otherwise return false.
- isFull() – Return true if the list is full, otherwise return false.

Stack as an ADT

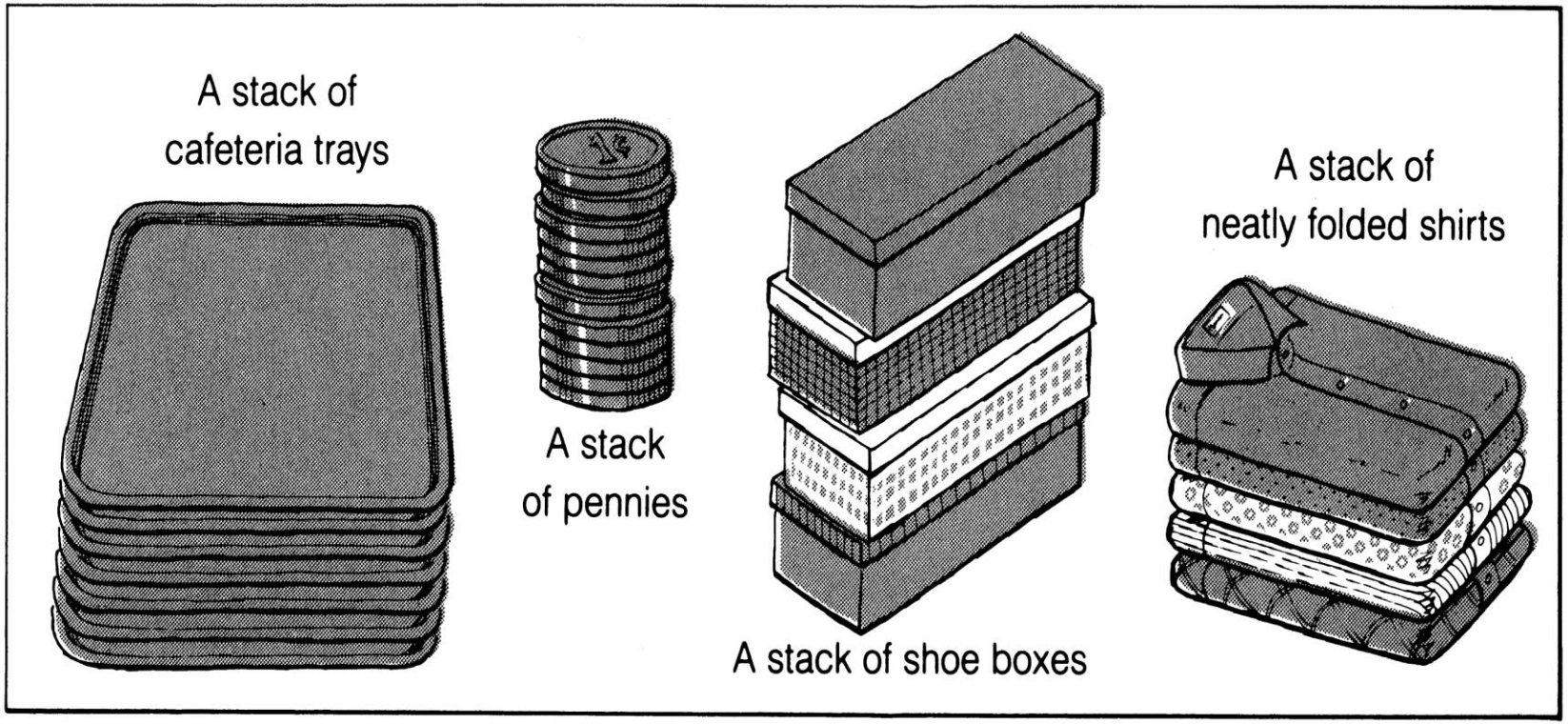
- When we define a **stack as an ADT**, then we are only interested in knowing the stack operations from the user's point of view.
- Means we are not interested in knowing the implementation detail at this moment.
- We are only interested in knowing what type of operations we can perform on the stack.

What is a stack?

- It is an ordered group of homogeneous items of elements.
- It is a linear data structure
- Elements are added to and removed from the top of the stack (the most recently added items are at the top of the stack).
- The last element to be added is the first to be removed (LIFO: Last In, First Out).

What is a stack?

- Example of stack



Stack Specification

- Definitions: (provided by the user)
  - MAX_ITEMS: Max number of items that might be on the stack
  - ItemType: Data type of the items on the stack

Stack Specification

- Stack operations:
  - push (data)
  - pop ()
  - top ()
  - peek ()
  - size ()
  - isEmpty()
  - isFull()

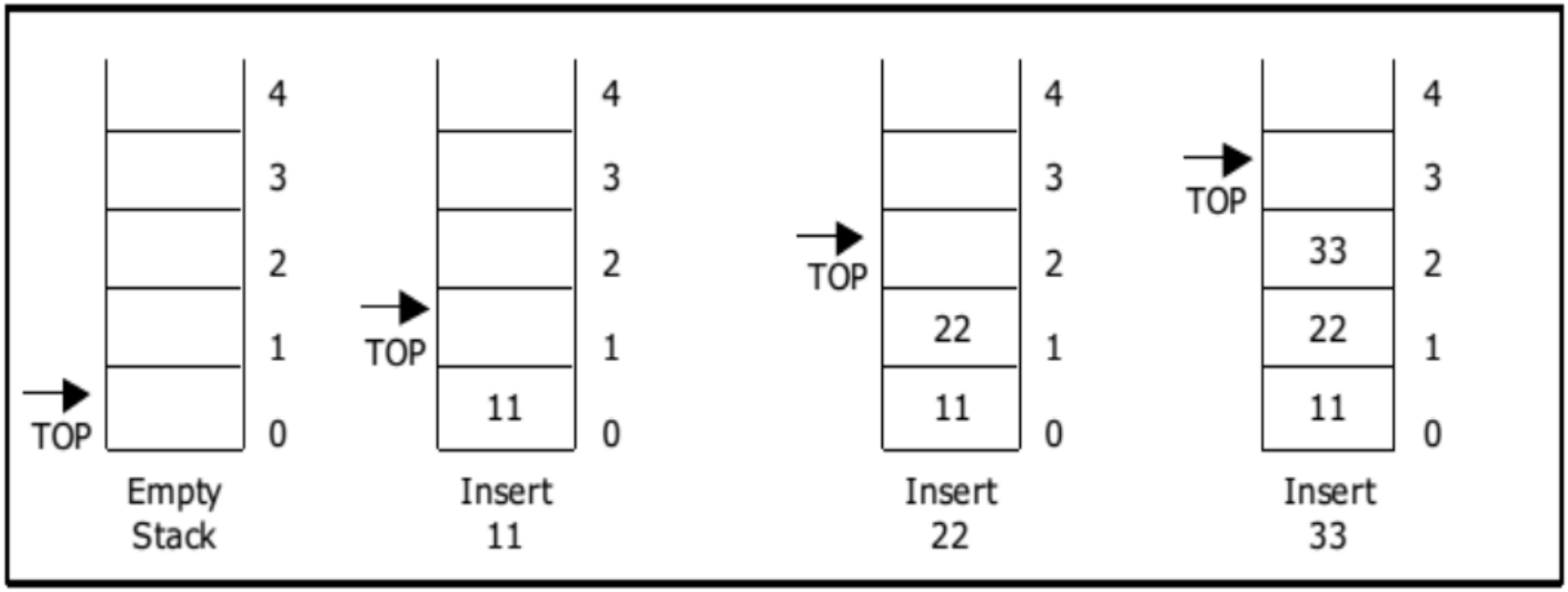
Push (ItemType newItem)

- Function: Adds newItem to the top of the stack.
- Preconditions: The stack has been initialized and is not full.
- Postconditions: newItem is at the top of the stack.

Pop (ItemType& item)

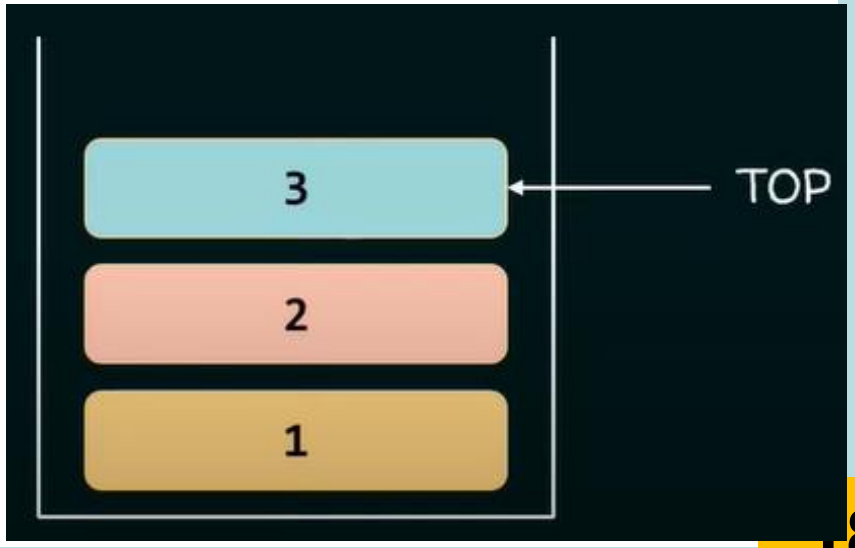
- Function: Removes topItem from stack and returns it in item.
- Preconditions: Stack has been initialized and is not empty.
- Postconditions: Top element has been removed from stack and item is a copy of the removed element.

Pop (ItemType& item)

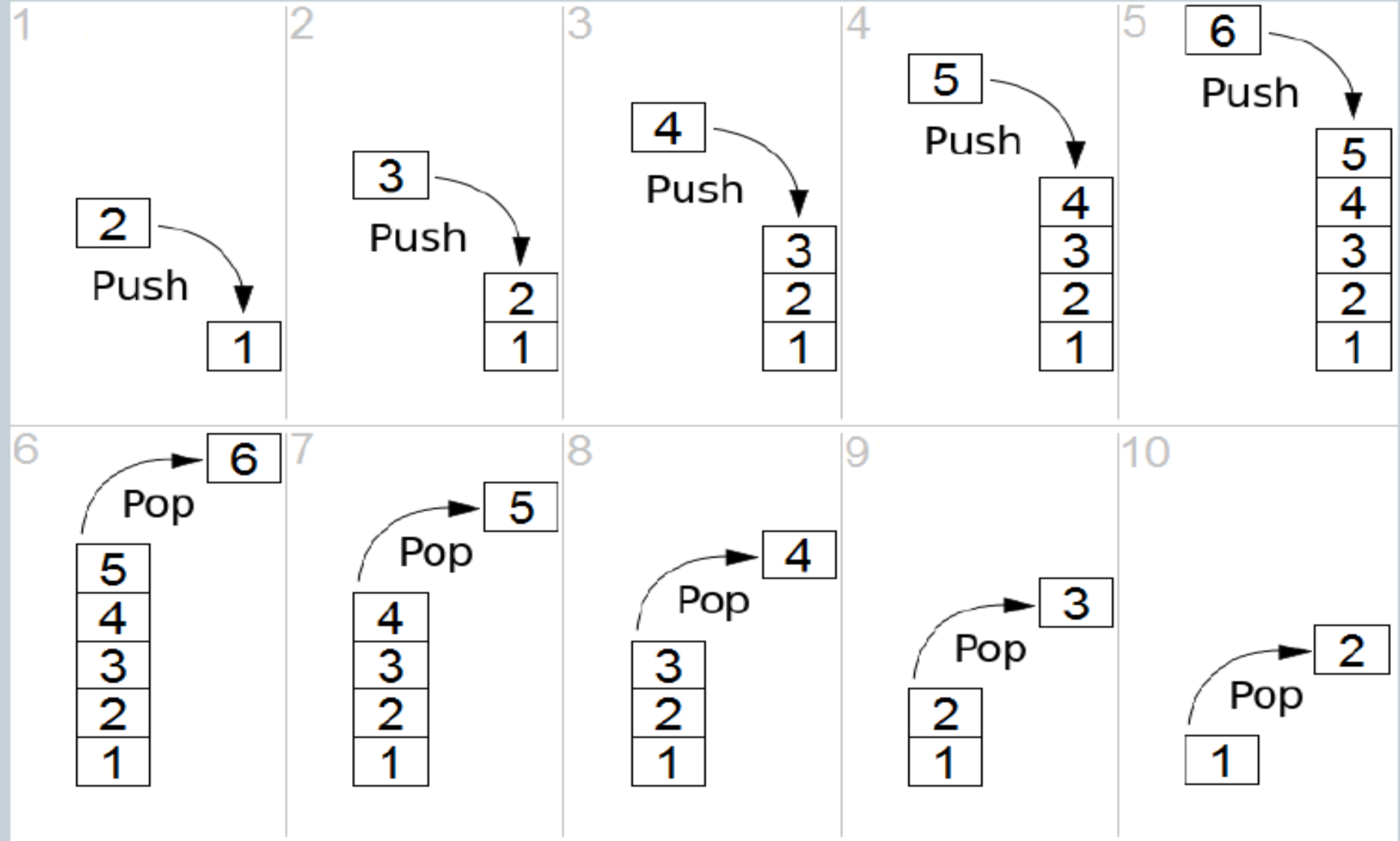


Stack: Operation

- Top() - Return the last inserted element without removing it
- Peek() - to read value from the top of the stack without removing it (Somewhere called top)
- size () - to return the number of items in the stack

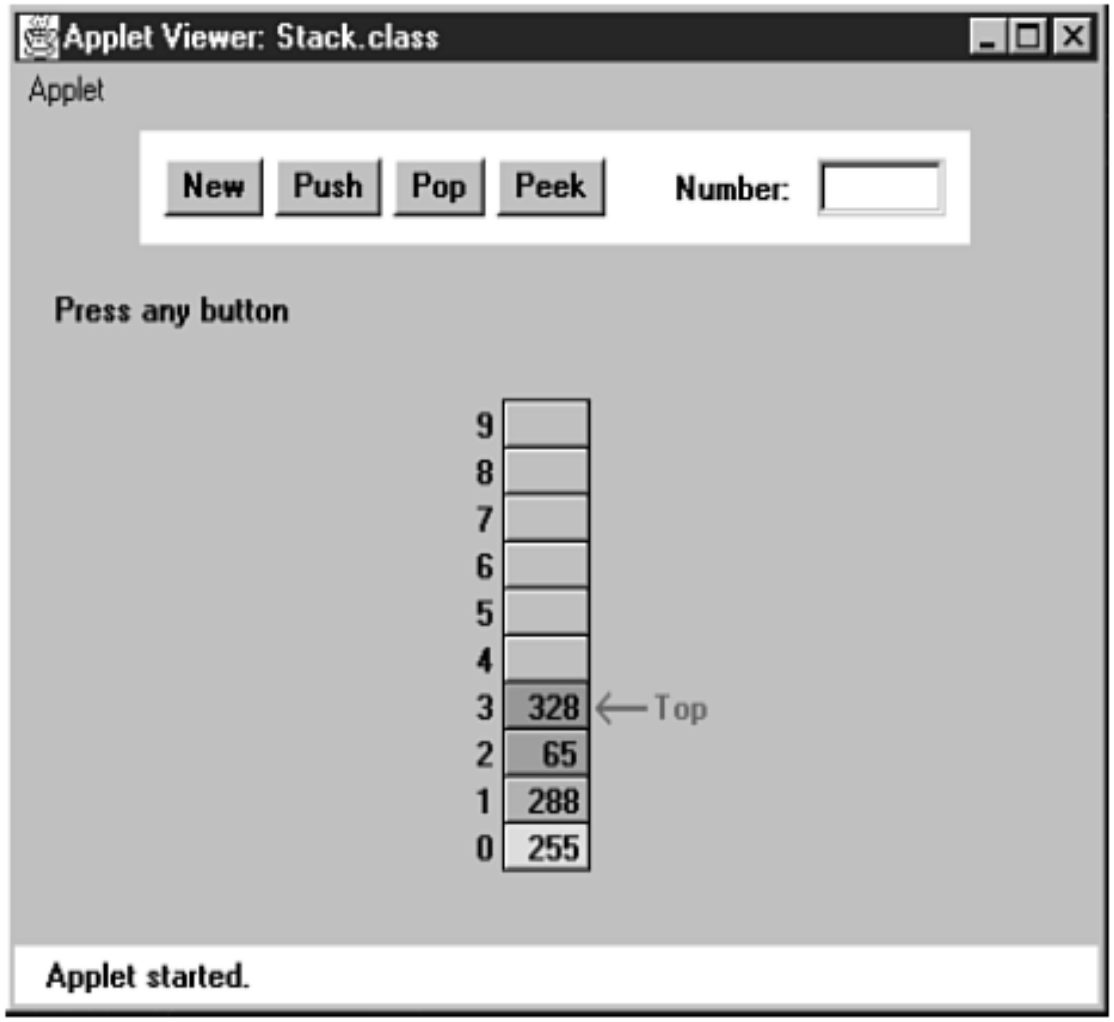


Stacks: Example of Push and Pop Operations



Stack example: stack operation

- The Stack Workshop applet.



Summary

- A stack allows access to the last item inserted, at the top of the stack.
- The important stack operations are pushing (inserting) an item onto the top of the stack and popping (removing) the item from the top.
- A stack is often helpful in parsing a string of characters, among other applications.
- A stack can be implemented with an array or with another mechanism, such as a linked list.

W6 – Lab 6

Exercise

- Create a class of Stack of anything (cafeteria trays, pennies, boxes, folded shirts) with the full operations:
  - push,
  - pop,
  - peek,
  - size,
  - isEmpty,
  - isFull

Thanks!