EC-200 Data Structures

LAB MANUAL # 06

**Course Instructor:** Lecturer Anum Abdul Salam

**Lab Engineer:** Lab Engineer Ayesha Batool

**Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Degree/ Syndicate: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Trait** | **Obtained Marks** | **Maximum Marks** |
| **R1** | **Application Functionality 20%** |  | 20 |
| **R2** | **Specification & Data structure implementation**  **30%** |  | 30 |
| **R3** | **Reusability**  **10%** |  | 10 |
| **R4** | **Input Validation**  **10%** |  | 10 |
| **R5** | **Efficiency**  **20%** |  | 20 |
| **R6** | **Delivery**  **10%** |  | 10 |
| **R7** | **Plagiarism above 80%** |  | 1 |
|  | **Total** |  | 10 |

**Total Marks = O**𝒃𝒕𝒂𝒊𝒏𝒆𝒅 𝑴𝒂𝒓𝒌𝒔 (∑6𝟏 𝑹𝒊 ∗ 𝑹7)

# Lab Manual # 06

# Implementation of Stack

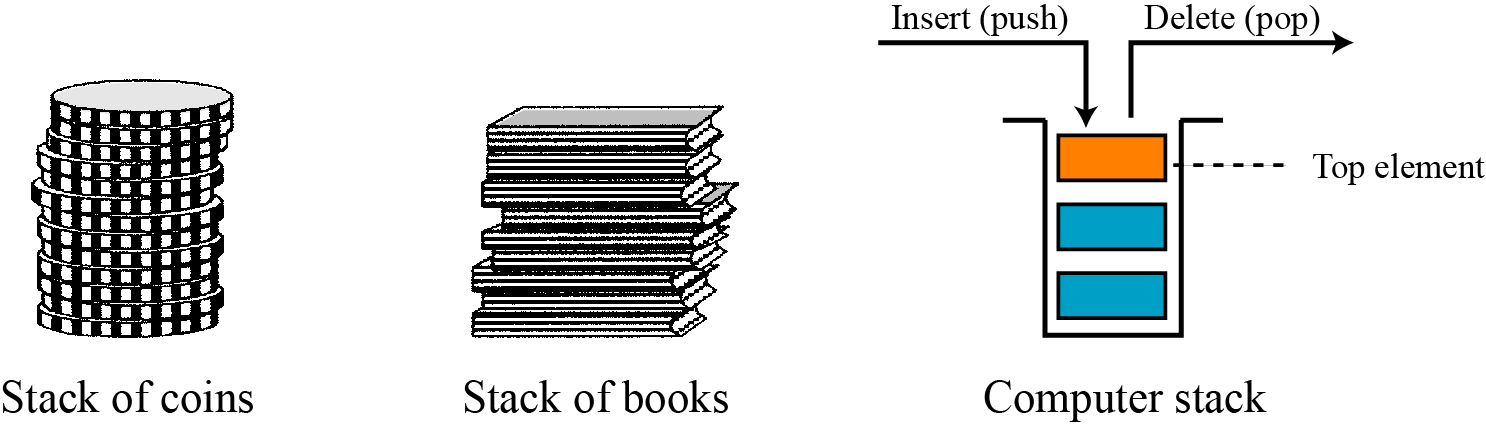
# Lab Objective:

To implement dynamic and static Stack ADT.

# Lab Description:

A stack is a restricted linear list. All additions and deletions are made at one end, the top. Stacks are known as last in, first out (LIFO) data structures. Stacks can be implemented using

1. Dynamic Approach (using lists)
2. Static Approach (using arrays)

**Figure 5.6:** Stack Representation

**Application of Stacks:**

1. Backtracking. (Function calls, Undo Redo operations)
2. Infix to postfix conversions

# LAB TASK # 01:

Design stack ADT using **static** approach and provide all the following functions.

1. Push (insert at start)
2. Pop (delete from start)
3. Retrieve Top (return top most element without deleting it)
4. Is Empty (to check is stack is empty)
5. Copy Constructor (for deep copying stack)
6. Destructor
7. Default constructor

## TEST PLAN:

**Execute your test plan. If you discover mistakes in your implementation of the List ADT, correct them and execute your test plan again.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.** | **Operations** | **Expected Results** | **Results/Status** |
| **1.** | Create an empty stack **myStack(size=5)** | Top=NULL |  |
| **2.** | Insert numbers 1-5 in stack |  |  |
| **3.** | Retrieve top from **myStack** | 5 |  |
| **4.** | Pop an element from stack | 5 |  |
| **5.** | Insert 6 in **myStack** |  |  |
| **6.** | Retrieve top from **myStack** | 6 |  |
| **8.** | Pop element from **myStack** |  |  |
| **9.** | Pop element from **myStack** |  |  |
| **10.** | Pop element from **myStack** |  |  |
| **11.** | Pop element from **myStack** |  |  |
| **12.** | Pop element from **myStack** |  |  |
| **13.** | Retrieve top from **myStack** | Stack empty |  |

# LAB TASK # 02:

Design stack ADT using **dynamic** approach and provide all the following functions.

1. Push (insert at start)
2. Pop (delete from start)
3. Retrieve Top (return top most element without deleting it)
4. Is Empty (to check is stack is empty)
5. Copy Constructor (for deep copying stack)
6. Destructor (De allocating stack memory)
7. Default constructor

## TEST PLAN:

**Execute your test plan. If you discover mistakes in your implementation of the List ADT, correct them and execute your test plan again.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.** | **Operations** | **Expected Results** | **Results/Status** |
| **1.** | Create an empty stack **myStack** | Top=NULL |  |
| **2.** | Insert numbers 1-5 in stack |  |  |
| **3.** | Retrieve top from **myStack** | 5 |  |
| **4.** | Pop an element from stack | 5 |  |
| **5.** | Insert 6 in **myStack** |  |  |
| **6.** | Retrieve top from **myStack** | 6 |  |
| **8.** | Pop element from **myStack** |  |  |
| **9.** | Pop element from **myStack** |  |  |
| **10.** | Pop element from **myStack** |  |  |
| **11.** | Pop element from **myStack** |  |  |
| **12.** | Pop element from **myStack** |  |  |
| **13.** | Retrieve top from **myStack** | Stack empty |  |

**THINK:**

1. What are the advantages of doubly linked list over singly linked list?
2. What are some applications of stacks?
3. In what scenario you’ll prefer using static stack implementation?
4. What is difference between implementation of static and dynamic stack in term of memory?