# University of Central Punjab Faculty of Information Technology

**Data Structures and Algorithms Spring 2024**

|  |  |  |
| --- | --- | --- |
| Graded Lab 1 | |  |
| **Topic** | * Abstract Classes * Templates * Stack * Stack Application |
| **Objective** | The basic purpose of this lab is to implement Stack, and test its Applications |

**Instructions:**

* Indent your code.
* Comment your code.
* Use meaningful variable names.
* Plan your code carefully on a piece of paper before you implement it.
* Name of the program should be same as the task name. i.e. the first program should be Task\_1.cpp

# void main() is not allowed. Use int main()

* **You have to work in multiple files. i.e separate .h and .cpp files**

# You are not allowed to use system("pause")

* **You are not allowed to use any built-in functions**

# You are required to follow the naming conventions as follow:

* + **Variables:** firstName; (no underscores allowed)
  + **Function:** getName(); (no underscores allowed)
  + **ClassName:** BankAccount (no underscores allowed)

# Students are required to complete the following tasks in lab timings.

**Task 1**

Create a C++ generic abstract class named as **Stack**, with the following:

**Attributes:**

1. Type\* stackArray;
2. int maxSize;
3. int stackTop;

**Functions:**

* virtual void Push(Type) = 0; Should add the element at the top of stack
* virtual Type Pop() = 0; Should remove the element from the top of stack

Using the above class **‘Stack’** make another derived **‘MyStack’.**

The subclass should include the following functions:

* **bool** [**isEmpty()**](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) : Returns whether the **Stack** is empty or not.
* **bool** [**isFull()**](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) **:** Returns whether the **Stack** is full or not.
* **int** [**size()**](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) : Returns the current size of the **Stack**.
* **Type** [**top ()**](https://www.geeksforgeeks.org/stack-top-c-stl/) : Returns the last element of the **Stack.**
* **Void display():** Should display the stack.

After Implementation of the functions in **MyStack** create menu based program to perform the following operations

1. Press 1 to remove duplicate elements in the stack.
2. Press 2 to reverse the stack
3. Press 0 to exit.
   * Write non-parameterized constructor for the above class.
   * Write Destructor for the above class.

revemeElement and reverseStack should be the member functions of the class ‘**MyStack’**

reverseStack take stack as an input and returns the reversed stack as an output, the function shoud reverse the input stack using only isFull, isEmpty, push, pop functions of the stack.

No other functions should be used.

**void reverseStack(MyStact<Type>& inputStack)**