University of Central Punjab

Faculty of Information Technology

Data Structures and Algorithms
Fall 2021

Lab 03	
Topic	• Abstract Classes • Templates • GrowAble Stacks
Objective	The basic purpose of this lab is to implement stack.

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)

Task 1

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

Attributes:

1. Type * arr;
2. int maxSize;
3. int currentSize;

Functions:

virtual void addElement (Type) = 0;

- Should add the element on the **List**

virtual void addElementAtFirstIndex(Type) = 0;

- Should add the element at the first position of the **List**

virtual void addElementAtLastIndex(Type) = 0;

- Should add the element at the last position of the **List**

virtual Type removeElement() = 0;

- Should remove the element from the current position of the **List**

virtual Type removeElementFromEnd() = 0;

- Should remove the element from the last position of the **List**

virtual void removeElementFromStart() = 0;

- Should remove the element from the first position of the **List**

- Write parameterized constructor with default arguments for the above class.
- Write Copy constructor for the above class.
- Write Destructor for the above class.

Task 2

Stack:

Implement Stack and perform the following operations. Your program needs to be menu based.

1. Press 1 to add a new item to the stack. **void push(Type)**
2. Press 2 to remove and return the last element from the stack. **Type pop()**
3. Press 3 to check if the stack is full. **bool full()**
4. Press 4 to check if the stack is empty. **bool empty()**
5. Press 5 to return the size of the stack. **int size()**
6. Press 6 to display the stack.
7. Press 0 to exit.