# University of Central Punjab Faculty of Information Technology

**Data Structures and Algorithms Spring 2024**

|  |  |  |
| --- | --- | --- |
| **Lab 05** | |  |
| **Topic** | * Abstract Classes * Templates * Stacks * Infix to Prefix, Infix to Postfix |
| **Objective** | The basic purpose of this lab is to implement Infix to Prefix conversion, Infix to Postfix conversion. |

**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. char\* stackArray;
2. int maxSize;
3. int stackTop;

**Functions:**

virtual void Push(char) = 0;

* + Should add the element at the top of stack virtual char Pop() = 0;
  + Should remove the element from the top of stack

# Task 2 Stack:

Implement a concrete subclass of **Stack** named **ExpressionConverter**, specifically designed for infix to prefix conversion and infix to postfix conversion. Inside the **ExpressionConverter** class, implement the **Push** and **Pop** functions to handle stack operations required for infix to postfix conversion. Develop the logic to convert an infix expression to its corresponding postfix form using the stack-based approach.

The subclass should include the following functions:

**bool** [**empty()**](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) : Returns whether the **Stack** is empty or not.

**bool** [**full()**](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**.

**char** [**top ()**](https://www.geeksforgeeks.org/stack-top-c-stl/) : Returns the last element of the **Stack.**

int precedence(char element);

**string convertToPostfix(const string& infix);**

**string convertToPrefix(const string& infix);**

# Implement both pure virtual functions Push () and pop() declared in base in ExpressionConverter

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

1. Press 1 to convert infix expression to prefix.
2. Press 2 to convert infix expression to postfix.
3. Press 0 to exit.
   * Write non-parameterized constructor for the above class.
   * Write Copy constructor for the above class.
   * Write Destructor for the above class.