



## Lab-12 Manual

*for*

### Introduction to Programming (CS200)

**Dr. Mian Muhammad Awais**

#### LAB GUIDELINES

- **Make sure you submit the lab before 11:50 AM.** Any late submission will not be graded afterwards. In case of internet connectivity or electricity issues make sure to email your assigned TA before 2:00 PM. **You should email your assigned TA ONLY.**
- For every lab, there will be a folder created on LMS. You must submit your work in the respective folder during the lab time, you and only you are responsible for your submissions.
- You will be allowed to discuss the questions in the first half of the lab session for a few questions. After that, there will be a portion of lab where you cannot converse and must work for yourself. No discussion is allowed in later time period.
- You should do your work with utmost clarity and precision. Do not waste your time trying to do something you do not understand. Ask Lab instructors for help, that is what they are there for.

- Any legitimate cheating case can and will be reported to Disciplinary Committee without any leniency. Plagiarism Software make our task easier.
- Please follow the lab etiquettes and follow code of conduct in the session.
- Do not start Personal chat during your zoom meeting and raise your hands before asking questions.
- **Make separate .cpp files for each question. Naming convention for .cpp files is: YourRollNumber\_TaskX.cpp. Before submission, copy all the .cpp files in a folder named LabX\_AssignedTAName\_YourRollNumber.zip. Submit the .zip file only! (no .rar file submissions). X should be replaced by appropriate number.** Failure to follow the naming convention may lead to deduction of marks.

## OBJECTIVES

- Stacks

## LAB EXERCISES

### Question # 1 [Marks: 100]

**Est. Time: 75 mins**

In this question, you have to implement a linked list based stack to make an infix expression calculator. **(Infix expression is the one where operators are written in-between their operands for example 10+12-5)**

You are required to do the following:

- Make your own linked list with appropriate functions you will be needing to implement a stack [20]
- Make your linked list based stack which supports the following functions: [30]
  - push() (adds an item at the top of the stack)
  - pop() (removes an item from the top of the stack)
  - peek() (shows the topmost item of the stack)
- Use your stack to implement infix expression calculator [50]

In int main(), you are required to prompt the user to enter an infix expression as a string (you can assume that the entered expression will never have a space in it so you do NOT have to write any code to cater for this scenario). Make appropriate declarations and calls **(you will have to push the operands and operators in the stack and then use the pop function to solve for the result of that expression)** and finally, print the result of the expression with an appropriate prompt.

### Sample Output # 1

Please enter an infix expression (no spaces): 7+3  
Answer: 10

### **Sample Output # 2**

Please enter an infix expression (no spaces): 7+3\*20  
Answer: 67

### **Sample Output # 3**

Please enter an infix expression (no spaces): 7+3\*20-50  
Answer: 17

**Goodbye and Goodluck!**