Lahore University of Management and Sciences

Lab-5 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

- Operator Overloading
- New and Delete in Operator overloading
- Getter Setter
- Dynamic Memory

LAB EXERCISES

Question # 1 [Marks: 40] Est. Time: 30 mins

Create a Class "**Date**" with following member variables
1. Month
2. Day
3. Year

• Write a function to print the Date in YYYY-MM-DD format.
• Write the code to subtract one Date from another & add one date to another via operator overloading and perform all the necessary error handling as well.
For simplicity assume every month is of 30 days and make sure that number of days should not exceed 30 and number of months should not exceed 12. In the case of year subtraction and addition perform the operation as shown.

Note: There are two separate operators so their will be two overloading functions i.e. + and – overloading.

Sample Output # 1

Sample Input:

First Date:

Month = 7

Day = 13

Year = 2020

Second Date:

Month = 4

Day = 9

Year = 2005

Sample Output:

Date Format: 2020-07-13

Subtract Date = 2015-03-04

Add Date = 2025-11-22

As you can see in the subtract part of the year we only subtracted the difference of the both years and vice versa for the addition.

Marks' distribution:

- Date in YYYY-MM-DD format: 5
- Code to subtract one Date from another & add one date to another via operator overloading and perform all the necessary error handling as well: 30
- Year Subtraction and Addition as shown in sample: 5

Question # 2 [Marks: 30] Est. Time: 40 mins

In this question you are to make a class "**ConcateName**" which has two data members:

- int **size**
- string* **name**

Second member is a pointer to a string variable which will be used to dynamically allocate a string array (name) of **size** where size will be prompted from user in main(). You have to make the following functions:

- Default constructor which initializes size and pointer data types with appropriate values

- Parametrized constructor that takes size as parameter and uses that to initialize the string array (name) dynamically
- Getters (tweaking them as per your requirements) for both the data members
- Overloaded >> operator which will be used in main() to take input for the class' objects
- Overloaded << operator to display the class' output in correct format from main()
- Overloaded + operator which concatenates arrays of 2 of the class' objects by concatenating them elementwise with space in-between. That is
  for all i: nameObj1[i] + " " + nameObj2[i]

In main(), you have to prompt the user for array size and using that, initialize 3 objects of the class so that all three of them have string arrays of same size. Take input into object1 and object2 of the class using overloaded >> operator. Then add object1 and object2 together using overloaded + operator and store the result in object3 and finally, using overloaded << operator display object3.

Sample output # 1:

Please enter size of your array: **8**

Object1 input:

Enter name for 1 position: **Irtasam**

Enter name for 2 position: **Ibrahim**

Enter name for 3 position: **Omair**

Enter name for 4 position: **Maham**

Enter name for 5 position: **Omer**

Enter name for 6 position: **Turyal**

Enter name for 7 position: **Rja**

Enter name for 8 position: **Maha**

Object2 input:

Enter name for 1 position: **Wains**

Enter name for 2 position: **Bukhari**

Enter name for 3 position: **Faqah**

Enter name for 4 position: **Ghazanfar**

Enter name for 5 position: **Ahmed**

Enter name for 6 position: **Neeshat**

Enter name for 7 position: **Batool**

Enter name for 8 position: **Sajid**

Object3 output:

Irtasam Wains

Ibrahim Bukhari

Omair Faqah

Maham Ghazanfar

Omer Ahmed

Turyal Neeshat

Rja Batool

Maha Sajid

Marks' distribution:

- Correct use of dynamic memory for array: 5
- All correctly overloaded operators: 15 (5 for each overloaded operator)
- Correctly working getters and constructors: 5
- Output format: 5

Question # 3 [Marks: 30] Est. Time: 40 mins

Make a class named Student and pass the name and age of student. You need to implement the constructor for Student class and a function Display to display name and age. Moreover, you need to implement the New and Delete operator inside Student Class, in order to delete the object at the end. New and delete operators are for creating dynamic objects/ allocating dynamic memory.

Note: You need to implement the statements mentioned in Sample Output in your code while making functions.

Sample Output # 1

Input:
Student name: Ali
Student Age: 20
Output:
Constructor Implemented
Student Name: Ali
Student Age: 20
Overloading delete operator

Marks' distribution:

Student Class implementation: 10
New & Delete Operator functions implemented in Class Student: 10
Display Function in Class Student with appropriate error handling: 5
Input/Output Format: 5

Best of Luck!