National University of Computer and Emerging Sciences

Lab Manual 06
Object Oriented Programming – CL1004

Course Instructor	MS. Anoosha
Lab Instructor(s)	Ms. Amna Zulfiqar Ms. Samman Ashraf
Section	BCS-9C
Semester	Summer 2023
Date	17-07-2023

Department of Computer Science
FAST-NU, Lahore, Pakistan

Lab Manual 06 – Classes with dynamic data members and pointers

Important Note:

- **You may find the syntax to accomplish these exercises from lecture demo.**
- **Add Necessary Comments in you code to justify your logic.**
- **Comment exercise number or statement at the start of your code**
- **Save each exercise in .cpp file with your roll no, ex and lab number e.g.**
- **22LXXXX_EX01_Lab01.cpp**
- **Upload cpp files on google classroom**
- **Make sure that the interface of your program is user friendly i.e. properly display information.**
- **Properly follow the coding standards.**

Objectives:

After performing this lab, students shall be able to:

- ✓ Create overloaded constructors
- ✓ Create destructors
- ✓ Handle dynamic memory for class data members as well as declaring objects.

Exercise:

In this Exercise, we are going to create a small scale **Event Management System**.

1. Create a class **Event** with following member variables:
   char* event_name, char* event_venue, char event_date[11] and char event_time[9].

   - Input format for event_date: dd-mm-yyyy
   - Input format for event_time: hh:mm am/pm
   - Event_name and even_venue will be dynamic arrays, size should not be more than number of characters +1 for null character.

2. Implement default constructor and overloaded constructor. Print "**Default Constructor Called**" and "**Overloaded Constructor Called**" in the respective constructors. The declaration for overloaded constructor is as follows:
   - Event(char event_name[20], char event_venue[50], char event_date[11],char event_time[9]);
   - This constructor will **deep copy** the input cstrings to member variables.

3. Implement all setters and getters for class Event. You can create a helper function userInput() to input event details.
   a. All setter functions should take input from user to change the value.
   b. When every new name is set for dynamic cstring member variables, old space should be deleted and new space should be allocated.

4. Implement the destructor ~Event() for class Event. Print "Destructor Called" in the destructor. Deallocate all the dynamically allocated memory of class data members.

In main, create event with following details.

event name: OOP LAB-06
event venue: Google Classroom
event date: 17-07-2023
event time: 05:00 PM

Call setEventName function to change the name to OOP LAB-QUIZ

Note:

- Deallocate all dynamically allocated memory.
- Do not use strcpy() function. Copy the character array manually where needed based on ending '\0' character