AIR UNIVERSITY	**Assignment No. 2** **Data Structures and Object-Oriented Programming** **Dr. Naveed Anwar Bhatti**	**Total Marks: 10 + 10** **Due Date: 16th Oct**

Instructions

Please read the following instructions carefully before submitting assignment:

It should be clear that your assignment will not get any credit if:

- **The assignment is submitted after due date.**
- **The submitted assignment does not open or file is corrupt.**
- **All types of plagiarism are strictly prohibited.**

Submission Procedure:

You can make project either in Microsoft Visual Studio or Dev-C++. Email me the CPP file. The name and the subject of email must be in the following format:

ASSIGNMENT1_SECTION_ROLL_FIRSTNAME

Example: ASSIGNMENT1_A_150600_NAVEED

Email ID: naveed.bhatti@mail.au.edu.pk

Objective

The objective of this assignment is to provide hands on experience of

- Classes in C++

Guidelines

- Code should be properly aligned and well commented.

Assignment

Problem Statement:

In this assignment, you will develop a ***University Building Management System***. The management system will mainly maintain the list of classrooms and labs with a proper menu system. Additionally, the project will also maintain the following parameters for each class and lab:

- Room Number;
- Building Number;
- Seating Capacity
- Courses allocated to each class room/lab

You have to create three classes:

```
class ClassRoom{};
class Lab{};
class University{};
```

ClassRoom class should have at least the following member functions with proper implementation:

```
ClassRoom();
~ClassRoom();

getSeatingCapacity();
getRoomNumber();
getBuildingNumber();
getNumber_of_Courses();

setSeatingCapacity();
setRoomNumber();
setBuildingNumber();
setCourses();

print_Course_Titles();
```

Lab class should have at least the following member functions with proper implementation:

```
Lab();
~Lab();

getSeatingCapacity();
getRoomNumber();
getBuildingNumber();
getNumber_of_Courses();

setSeatingCapacity();
setRoomNumber();
setBuildingNumber();
setCourses();

print_Course_Titles();
```

Finally, **University** class should have at least the following member functions with proper implementation:

```
University();
~University();
getTotalClassRooms();
getTotalLabs();
setTotalClassRooms();
setTotalLabs();
```

Important rules to follow:

- For each of the member functions set an appropriate number of input arguments and return type.
- You need to dynamically create the number of class rooms/labs based on user input. The same rule applies to the number of courses allocated to each classroom/lab.
- Everything needs to be done via ***University*** object. For this, you will make two pointer objects of ***ClassRoom*** and ***Lab*** class as data members of ***University*** class.
- Program should have a proper menu system.
- You need to delete the dynamically created objects properly via destructor.

Bonus Marks (10):

- Bonus marks will be given if you will add more functionality to each of the class (more data member, more function members). For example, besides just taking the course titles allocated to each classroom/lab as an input from the user, if you will also get the end/start timings of each course from the user and print the time table for each class, you will get bonus marks.
- Mention each of the additional function clearly in the code via commenting

Deadline

Your assignment must be emailed on or before 16th Oct.