



King Hussein School for Computing Sciences
Department of Computing Science
Object Oriented Programming Lab
Assignment #7

Instructor: Bushra Alhijawi

Fall 2020/2021

Student Name	
Student Univ. ID	
Date	

	Q1 (50)	Q2 (50)
100%		
90%		
75%		
50%		
0%		

Total (100):_____

Grading Criteria

- **100%:** Program is fully correct, code is well-indented, identifiers are well-named and the output is well formatted.
- **90%:** Program is fully correct but meaningless identifier names are used, indentation is bad, or output is not well formatted.
- **75%:** Program runs mainly correctly. It fails to adhere to some of the specifications.
- **50%:** Program contains syntax errors but seems correct.
- **0%:** Program has serious syntax and/or logical errors.

Lab Objectives

- Practice using constant member functions and constant data members.
- Practice using static member functions and static data members.
- Practice using constant object.

Lab Instructions

- Create a new folder on desktop and name it by your "ID-FirstName-LastName-Lab1". Make sure to save your solutions in this folder.
- You must upload your solutions to e-learning as follows:
 - **Each project** you create it as a .zip file.
 - A **text file** associated with each project contains a copy of the code.

Lab Exercises

This assignment consists of two exercises. You should submit your code in **three files** on the e-learning: one header file (.h), two CPP source files (.cpp). Also, in your header file, you must prevent multiple file inclusion. Your code should use separate implementation of the class.

Exercise 1 – Conference Class

Exercise Objectives

- ✓ Define class
- ✓ Using constructor and destructor
- ✓ Using constant and static class members

Problem Description

In this exercise you will develop the two file in this exercise called **conference.h** and **conference.cpp**. **conference.h** contains the class declaration and **conference.cpp** contains the class implementation.

➤ A class called conference (as shown in the class diagram) contains:

- Four private variables:
confId (int), confType (String), noOfSeats (int) and numOfConf (int). confType can be either “Academic” or “Entertainment”. numOfCars is a static **counter** that should be
 - Incremented whenever a new conference object is created.
 - Decrementd whenever a conference object is destructed.
- Two constructors (parametrized and copy constructor) and a destructor. Initialize the confId using numofConf. Whenever the copy constructor is called, it should print “Copy Constructor is called”
- Getters and setters for the conference type, ID, and date. And static getter function for numOfConf.
- All member functions that do not modify data members should be **constant**.
- A function **print()** that prints the details of object.

conference
- confId: const int - confType: String - noOfSeats: int - numOfConf : static int
+ conference(t:string, s:int) + conference (& c: conference) + ~ conference () + getConfId():int + getConfType:String + getNoS ():int + setConfId(i:int) + setConfType(T:String) + setCNoS(s:int) + getnumOfConf (): static int + print(): void

Exercise 2 – Driver Program

Exercise Objectives

- ✓ Declare objects
- ✓ Call functions

Problem Description

Write the **main** function that uses **conference** class as follows:

- Print out the number of available conference.
- Declare 4 objects of type **conferences**. Ask the user to enter the details of these objects.
- Print out the details of the four objects.
- Print out the conference ID of which has a larger number of seats.
- Print out the number of available conference.
- Add a new non-member function called `newConference` that receive the second object as parameter. Use the Pseudocode below to write the function body:

```
void newConference(ob: conference){  
    Declare new object of type conference as a copy of ob.  
    Print the number of available conference.  
}
```
- Call the function in the main.
- Print out the number of available conference.
- Define a constant object of type **conferences**.
- In a comment section, for each of the member functions in class **conferences**, list which functions can be called on the constant object and which cannot. Justify your answer.