PUNJAB UNIVERSITY COLLEGE OF INFORMATION TECHNOLOGY

MID-TERM EVALAUTION

TOTAL MARKS = 50

Course: Object Oriented Programming Lab
Semester: Special Section BS (CS & IT)
Deadline: 9th May 2022

Instructions:

Note: Violation of any of the following instructions may lead to cancellation of your submission.

- Create a folder and name it by your student ID like (BCSF20M550 / BITF20439)
- Paste the .cpp file for each question with the names such as Q1.cpp, Q2.cpp and so on into that folder**.**
- If any student fall in copy of assignment, in case of **PLAGIARISM** both students marked assignment as Zero.
- **Email the Zipped folder (aleem.subhani@pucit.edu.pk).**

Question # 1:

Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay a 50-cent toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by, and of the total amount of money collected.
Model this tollbooth with a class called tollBooth. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both of these to 0. A member function called payingCar () increments the car total and adds 0.50 to the cash total. Another function, called nopayCar (), increments the car total but adds nothing to the cash total. Finally, a member function called display () displays the two totals. Make appropriate member functions const.
Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the Esc key should cause the program to print out the total cars and total cash and then exit.

Question # 2:

Create a class called time that has separate int member data for hours, minutes, and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. Another member function should display it, in 11:59:59 format. The final member function should add two objects of type time passed as arguments. A main () program should create two initialized time objects (should they be const?) and one that isn't initialized. Then it should add the two initialized values together, leaving the result in the third time variable. Finally, it should display the value of this third variable. Make appropriate member functions const.

Instructor: Muhammad Aleem Subhani

Question # 3:

In ocean navigation, locations are measured in degrees and minutes of latitude and longitude. Thus, if you're lying off the mouth of Papeete Harbor in Tahiti, your location is 149 degrees 34.8 minutes west longitude, and 17 degrees 31.5 minutes south latitude. This is Chapter 6 260 written as 149°34.8' W, 17°31.5' S. There are 60 minutes in a degree. (An older system also divided a minute into 60 seconds, but the modern approach is to use decimal minutes instead.) Longitude is measured from 0 to 180 degrees, east or west from Greenwich, England, to the international dateline in the Pacific. Latitude is measured from 0 to 90 degrees, north or south from the equator to the poles.

Create a class angle that includes three member variables: an int for degrees, a float for minutes, and a char for the direction letter (N, S, E, or W). This class can hold either a latitude variable or a longitude variable. Write one member function to obtain an angle value (in degrees and minutes) and a direction from the user, and a second to display the angle value in 179°59.9' E format. Also write a three-argument constructor. Write a main () program that displays an angle initialized with the constructor, and then, within a loop, allows the user to input any angle value, and then displays the value. You can use the hex character constant '\xF8', which usually prints a degree (°) symbol.

Question # 4:

Create a class that includes a data member that holds a "serial number" for each object created from the class. That is, the first object created will be numbered 1, the second 2, and so on.

To do this, you'll need another data member that records a count of how many objects have been created so far. (This member should apply to the class as a whole; not to individual objects. What keyword specifies this?) Then, as each object is created, its constructor can examine this count member variable to determine the appropriate serial number for the new object.

Add a member function that permits an object to report its own serial number. Then write a main () program that creates three objects and queries each one about its serial number. They should respond I am object number 2, and so on.

Question # 5:

Note that one advantage of the OOP approach is that an entire class can be used, without modification, in a different program. Use the fraction class from Exercise 11 in a program that generates a multiplication table for fractions. Let the user input a denominator, and then generate all combinations of two such fractions that are between 0 and 1, and multiply them together. Here's an example of the output if the denominator is:

```
                1/6     1/3     1/2     2/3     5/6
---------------------------------------------------------------------------------
1/6             1/36    1/18    1/12    1/9     5/36
1/3             1/18    1/9     1/6     2/9     5/18
1/2             1/12    1/6     1/4     1/3     5/12
2/3             1/9     2/9     1/3     4/9     5/9
5/6             5/36    5/18    5/12    5/9     25/36
```

Question # 6:

Write a class player that contains attributes for the players name, average and team. Write three functions to inputs change and display these attributes also write a constructor that ask for input to initialize all the attributes.

Question# 7:
Define a class for a bank account that include the following data members

- Name of the depositor
- Account number
- Type of account
- Balance amount in the account

The class also contains the following member function.

- A constructor to assign initial values
- Deposit function to deposit the sum amount, it should accept the amount as parameter.
- Withdraw function to withdraw an amount after checking the balance. It should accept amount as parameter.
- Display function to display the name and balance.

Question# 8:
Write a class Run that contain the following data members.

- The name of the runner
- The distance cover by a runner

The class has following member functions.

- get () function to input runner name and distance.
- Show () function to display the runner's name and distance.

The user should be able to show the name of the runner who ha covered the longest distance at any time of that point.

Question: 9:
Write a class that contains the following attributes.

- The name of car
- The direction of car (**E, W, N, S**)
- The position of car (from imaginary zero point)

The class has the following member functions.

- A constructor to initialize the attributes
- Turn () function to change the direction of car to one step right side (e.g., if the direction is **E,** it should be change to **S** and so on).
- Overload the Turn () function to change the direction to any side directly. It should accept the direction as parameter.
- Move () function to change the position of car away from zero point. It should accept the distance as parameter.

Question# 10:
Write a class Travel that has the attributes Kilometers and hours. A constructor with no parameter initializes both data members to 0. A member function gets () function inputs the values and show () function displays the values. It has a member function add () that takes an object of type Travel, adds the kilometers and hours of calling object and the parameter and returns an object with added values.

Note: All member functions define outside the class.

Instructor: Muhammad Aleem Subhani