National University of Computer & Emerging Sciences, Karachi
Spring-2023 School of Computing (BSCS, BSSE, BSCY, BSAI)

Assignment # 1

Subject: Object Oriented Programming - CS1004	**Post Date: 10/2/2023**
Total Marks: 50	**Due Date: 19/2/2023, 11:59 PM**
Course Instructors: Dr. Farooque Hassan Kumbhar, Dr. Abdul Aziz, Mr. Zain ul Hassan, Ms. Sobia Iftikhar, Ms. Eman Shahid, Ms. Abeer Gauher, Mr. Basit Ali, Ms. Javeria Farooq, Ms. Sumaiyah Zahid, Mr. Shahroz Bakht, Ms. Abeeha Sattar	

Instructions to be strictly followed.

- Each student should submit these files:
  - ***A zip of all source files named as "A1-Q#[StudentID]" where # is the question number and Student ID is your ID.***
  - ***A DOC file where they copy code for each question and screen shot of the output. This document contains all the questions, answer codes and output in sequence. Name this document as "A1-[StudentID].docx".***
  - ***All the submissions will be made on Google Classroom.***
- Each output should have STUDENT ID and NAME of the student at the top.
- It should be clear that your assignment would not get any credit if the assignment is submitted after the due date.
- Zero grade for plagiarism (copy/ cheating) and late submissions.

Question One. **Total Marks /10**

As a programmer, you are tasked to develop an object-oriented simulation of a heating system in a smart home using the "Room" and "Radiator" classes.

The "Radiator" class should have the following features: **Marks /4**

- RadiatorID and isOn member variables.
- A default constructor that assigns a unique RadiatorID value (incremented by 15 for each new object) with starting value on the last 3 digits of the student ID "22K – 1345" (e.g. 345). **You are required to input your student ID and extract the digits.**
- The radiator starts in the off state.
- An accessor method for RadiatorID.
- A public method called "heats" that links the Room object with the Radiator object.

The "Room" class should have the following features: **Marks /4**

- roomName, seatingCapacity, and numRadiators member variables.
- A constructor that accepts roomName and sets seatingCapacity to a default value of 12.
- A public method "isHeatedBy" that links the Radiator object with the Room object and returns a message indicating the status of the radiator integration. This method returns a String containing one of the following messages indicating whether the Radiator was successfully added or not:
    - "Radiator already added to room."
    - "Radiator successfully added to room."
    - "Cannot add Radiator. Room has a maximum number of radiators."
- A Room can be heated by 0 to 2 unique radiators.

In the main program do the following: **Marks /2**

Create a "Room" object "room" and two "Radiator" objects "rd1" and "rd2" and link them such that the "Room" object "room" is heated by both "Radiator" objects.

Question Two. **Total Marks /10**

A data science company is trying to make a platform **DataHub** (such as StackOverflow) where different data scientists can post their coding problems related to data science only. Other data scientists can help in solving their problems. You are hired as an initial developer and your task is to create and object-oriented paradigm for the **DataHub** in the following manner:

DataScientist Class: **Marks /5**

Your application must keep track of Data Scientist details including their *first name, last name, highest education, age, country, number of answers given, and number of questions asked*.
Every user is going to have a unique ID which is issued automatically on their object creation, and the generated ID will be used throughout the program for that particular object. **The ID will be generated using the second and the last digit of your student ID. You are required to input your student ID and extract the digits.** The class must have:

- A parameterized constructor that initializes the value of every Data Scientist user.
- These data members are read-only: number of answers given and number of questions asked.
- These data members can be accessed or updated at any given time: first name, last name, age, and country.
- You may set the highest education but can't access it.
- The class must have a member function **AskQuestion** in which any given problem is posted by the user and it should increase the count of the number of questions asked by that user automatically.
- The class must have a member function **AnswerProblem** in which the user solved any queries posted by others and it should increase the count of the number of answers given by that user automatically.

[Please note posting questions and posting answers are just used to increase the count right now. No mechanisms are required for saving those questions or answers.]

Admin Class: **Marks /3**

Admin details include their *first name, last name, age, country, and admin ID*.
Admin class also keeps track of every new user created. A new user creation must increment the user count known by only Admin Class. Admin class must have

- A parameterized constructor that initializes the value of admin and also assigns AdminID.
- These data members can be accessed or updated at any given time: first name, last name, age, and country.
- The class must have a member function **Total User.** Calling this function will show the total count of data scientists created but only the admin knows the total count.

In the main program do the following: **Marks /2**

Implement the above scenario by creating objects. Show the functionalities of both the classes by invoking the functions appropriately.

Question Three. **Total Marks /10**

A smartwatch is a wearable computer in the form of a watch; modern smartwatches provide a local touchscreen interface for daily use and calculate your health essentials.

While exercising, you can use a heart-rate monitor installed in your smartwatch to see that your heart rate stays within a safe range.

The formula for calculating your maximum heart rate in *beats per minute is 220 minus your age in years*. Your target heart rate is a range that is 50–85% of your maximum heart rate.

Your task is to create a class called **HeartRates**.

Marks /2

- The class attributes should include the person's first name, a unique ID, and date of birth (consisting of separate attributes for the month, day, and year of birth). **The ID will be the third digit of your student ID. You are required to input your student ID and extract the third digit.**
- Your class should have a constructor that receives this data as a parameter. For each attribute, provide set and get functions.

The class has the following functions: **Marks /6**

  - a function getAge that calculates and returns the person's age (in years),
  - a function getMaxiumumHeartRate that calculates and returns the person's maximum heart rate
  - a function getTargetHeartRate that calculates and returns the person's target heart rate.

Marks /2

- Write an application that prompts for the person's information, instantiates an object of the class HeartRates, and prints the information from that object including the person's first name, and date of birth—then calculates and prints the person's age in (years), maximum heart rate and target-heart-rate range.

Question Four. **Total Marks /10**

Atrium Cinemas need to automate their system to keep track of the tickets sold for the movies that are screened at the cinema halls.

Create a class named as **"Movie"**. **Marks /3**

- The class contains member variables that include movie name and showtime.
- Users can only see which movies are being shown and at what times. Include member functions that display what movies are being shown and their showtimes.

Create a class named as **"Ticket"**. **Marks /4**

- The class contains member variables that include the row number, seat number, ticketID, and a boolean / flag variable that maintains the status of whether the seat has been sold or not.Status (boolean / flag) is a private member variable.
- The ticketID will be the first two digits of your studentID. Generate unique integer values for the ticketID using the first 2 digits of your student ID "22K – 1345" so 13 would be considered as the ticketID. **You are required to input your student ID and extract the digits.**
- Include member functions that check if the ticket has been sold, update the ticket status to sold and a display function that displays the sold status, row number, ticketID and the seat number for the sold ticket.

In the main program do the following: **Marks /3**

Implement the above scenario by creating objects. Show the functionalities of both the classes where a user can buy a ticket for a particular movie. Additional functions and variables can be added as needed.

Question Five. **Total Marks /10**

Let's go over a very simple scenario:

Scenario: **<u>An application dedicated to providing the right shoes for its users</u>**

(But we'll skip the searching part for now)

Let's consider the **User** class for the application. Each person(user) can be represented by a specific set of attributes, for example, their user ID, their name, their age, their height, their gender, their shoe size, etc. Each person has some actions they can perform on the application. One such function is notifying whether the user is an infant, toddler, child, teenager or adult (We'll assume that the parents handle the accounts for children under 13). Furthermore, the user attributes can be accessed and updated by the application by some functionality provided by the user class.

Similarly, a **Shoe** can be represented by its size, width, style (running shoes, sneakers, stiletto, etc.), brand, color and demographic (infant, toddler, child, adult). Similar to the user class, the attributes for any shoe can be accessed and updated by the functionality provided by the shoe class.

Your task for this question is to create an application that asks the user to enter their information. Your program should create an object for the user and store the information inside the object using the functionality that you have created for these classes.

Afterwards, your program should ask the user about which type of shoes they want to purchase. Once again, after you have taken the input from the user, you should create an object for the shoe class, and store values in that object.

Your program should have the following checks for ages, when assigning the value to the demographic attribute in the shoe object:

- infant: ages 0 -2 years (0 assumes that infant is some months old)
- toddler: 3-5 years
- child: 10-12 years
- teenager: 13-19 years
- adult: 19+ years

Following things are required for this question:

1. **Marks /2**

User and Shoe classes (along with all of the attributes and functions). User ID should be an integer array of length 2. It should hold your roll number within its indices in this manner: If your roll number is 12k-2034, it should be stored in the array as: {12, 2034}

2. **Marks /2**

Default and Parameterized constructors for these classes

3. **Marks /1**

Accessor and Mutator functions for these classes

For questions 4-6, provide a menu-driven interface for the user.

4. **Marks /2**

Objects for User and Shoe should be created in your main function. This should be done by using user provided values.

5. **Marks /2**

Users should be allowed to update the information that they have provided as many times as they would like. Values should always be updated using the setter functions.

6. **Marks /1**

Users should be allowed to view the updated details as many times as they would like. Values should always be retrieved by using the getter functions.

7. **Marks /???**

Bonus(?): Create a **global** function that accepts a User and Shoe Object and displays all their information by using the getter functions.