Lab 2: Basic class design and constructors

Total marks: 30 marks (3x10)

Deadline: same day before 11:59 pm (sharp)

Instructions: (MUST READ)

- **No compensation or makeup lab.**
- **Don't discuss with peers. Changing variable names/ changing for to while loop will not help you in hiding cheating attempt!**
- **You are not allowed to ask TA to verify/ prove your cheating case! Any such complaint from TAs will result in serious consequences. Don't expect any positive response from TAs in such regard.**
- **Cheating cases will be given ZERO first time. Second attempt to cheat will result in deduction in sessionals. I'll report the regular cheating attempts to Discipline committee without any prior warning.**
- **You are not allowed to consult Internet. Plagiarism cases will be strictly dealt.**
- **You can ask relevant queries (in case you are stuck somewhere or a very VALID query regarding some error) from TAs only between 9 am – 12pm slot.**
- **You are not allowed to ask TAs about problem statement. Understanding the problem statement is a part of that question. Do whatever you are asked and what you understand.**
- **You must submit the lab solution BEFORE 12pm. Even a few minutes late submissions will not be considered. Make sure to do proper management of time/Internet connectivity/power failure or whatever issue is possible!**

General instructions for all tasks: (marks will be deducted if the instructions are violated)

Note: All the programs should be implemented using class. You can take input in main() function and then call appropriate methods/ member functions of a designed class to set and get values. No need to perform the following lab tasks in multiple files. You can implement all logic in one cpp file.

• The attributes of class should be declared as **private** and member functions as **public**.

• All the member functions (expect constructor) should be declared inside the class and defined outside the class.

• You should not initialize the attributes while declaring them in class. The values should be assigned using member functions only. E.g. you cannot declare like:

```
Class Person
{
Private:
int age=25;
}
```

• The values should be initialized using a constructor. There must be a constructor in your defined class.

• All inputs should be taken in *main()* and all the final results should also be reported/ displayed in the main function.

• All the logic should be implemented in class' member functions. Main() should only input and output relevant values by calling relevant functions of the class.

TASK 1:

Assume you have to write a program for a cricket game. There are two teams and each team has a value for score and number of wickets. You are required to design a class **Team** that has relevant member functions for setting the score and wicket's value and for getting these values too.

(a). The score and wicket of both the teams should be taken as an input from the user in **main().** The main function should interact with the class **Team** to set and get values.

Expected output (console):

```
****Enter data for Team1****
Enter score: 24
Enter the number of wickets: 4
Team 1 has score: 24 and number of wickets: 4
****Enter data for Team2****
Enter score: 67
Enter the number of wickets: 9
Team 1 has score: 67 and number of wickets: 9
```

Main function (driver function) should do the following tasks:

```
//declare objects to interact with the class
//take input from user
//call set methods of class
//call getter methods and print to the console
```

(b). Declare two more objects and initialize one of the object using default constructor and the other using parameterized constructor. You may send hardcoded values while calling parametrized constructor.

Expected output (console):

```
**** Team3 created (default)****
Team 3 has score: 0 and number of wickets: 10
**** Team4 created (parameterized)****
Team 4 has score: 11 and number of wickets: 2
```

Main function (driver function) should do the following tasks:

```
//declare objects to interact with the class
//call getter methods and print to the console
```

(c). In **main(),** write a logic that finds the team with 1) minimum score , 2) maximum score, 3) with greater than 6 wickets**.** The main function should interact with the class **Team** to set and get values. First display the scores and wickets for all teams and then report 1,2, and 3.

Expected output (console):

Team 1 has score: 24 and number of wickets: 4
Team 2 has score: 12 and number of wickets: 9
Team 3 has score: 120 and number of wickets: 3
Team 4 has score: 11 and number of wickets: 7

Team 3 has maximum score
Team 4 has minimum score
Team 2 and 4 have greater than 6 wickets

Main function (driver function) should do the following tasks:

```
//continuing the existing program…
//display values for teams using getter methods
//find maximum etc..as asked..
```

TASK 2:

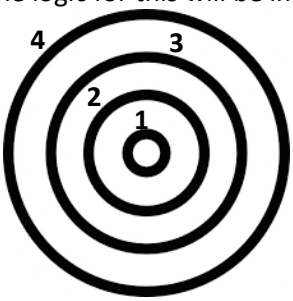
Design a class of Circle with atleast two data members: radius, originx, and originy. Origin is a central point of a circle and radius is the distance between central point and boundary of a circle. Since origin is a point therefore it can be modelled with x and y coordinates.

(a). Write a program that finds the area, diameter, and circumference of a circle (create one sample object and find the following) by creating appropriate class. The area, circumference, and diameter can be calculated using following formulas, respectively:

*Area=pie*radius*radius*
*Circumference=2*pie*radius*
Diameter=radius/2

All the inputs should be taken from the user in main function to set data members of class (like done in Task 1) and then relevant member functions of the class should be called get values and find *area, circumference, diameter*. The functions for finding them must be a part of class. Main function should only call relevant methods.

(b) Create 4 circles and initialize them using parameterized constructor (you can hardcode values while calling constructor). Find and display which of the circles are **concentric**. That is, circles having the same central point (as shown below). (Hint: the logic for this will be implemented in main())



(c) For the identified concentric circles, find the label for each circle. Label is the position number (as shown above; most internal circle has label 1, and so on.. At the end, also display the labels that you assigned. (Hint: the logic for this will be implemented in main())

(d) Find and display: which of the concentric circle has **diameter greater than 12**. You must differentiate the relationship between diameter and radius of any circle. (Hint: the logic for this will be implemented in main())

Purposefully, I haven't attached sample output and structure of main function. Please create a simple example scenarios (dry run; e.g. like I have considered four circles in picture above). Do processing first on paper and design structure/logic, then go for implementation. The requirements (what I want from your code) are clearly mentioned above. So, do what is logical. Understanding and designing structure of a program is a part of this question.

TASK 3:

Consider a class named Job that has deadline as a data member and relevant getter/setter method(s). Assume you have to schedule two most earliest jobs on the basis of their deadlines. That is, if there are three jobs in the system with deadlines (deadline1, deadline2, and deadline3, respectively) then the system should report the top two earliest jobs (with smallest *deadline* value). You might need to find the *deadline* with smallest and second most smallest value.

Expected output (console):

Job 1 has deadline 12
Job 2 has deadline 1
Job 3 has deadline 11
Job 4 has deadline 45
Job 5 has deadline 6
Job 6 has deadline 9
Job 7 has deadline 66
Job 8 has deadline 5
Job 9 has deadline 27
Job 10 has deadline 34
Job 2 has earliest and job 8 has second most earliest deadline.

You need to do following tasks in main()

//Create 10 objects of class and initialize them with hardcoded values using parameterized constructor.
//Display the jobs with deadlines using getter method
//Find and report the two most earliest jobs

Extra task, just for practice (neither graded nor a bonus)

Implement a class *RandomNumberGenerator* that has *number* as a data member. It has two member functions named as *int GetRandom10 ();* and *int GetRandom30 ();*. First one generates random number between 0-10. Second one generates random number between 0-30. Main function should call appropriate functions of class to generate 5 random numbers lying in the range (0-10) and 5 random numbers lying in the range (0-30). This should all be done by calling the functions of the class

RandomNumberGenerator. The program should then display in main() the maximum number out of these 10 generated random numbers.

Hint: You can store the 10 generated random numbers in an array and then find maximum value from this array.

Expected output (console):

Random numbers generated in the range (0-10):
2
4
6
1
1
Random numbers generated in the range (0-30):
5
22
11
9
10
Maximum number: 22