**King Saud University**
**College of Computer & Information Science**
**CSC111 – Assignment 9**
**All Sections**
---

## Question 1

Design a class named **Fan** to represent a fan. The class contains:

- Three constants named **SLOW**, **MEDIUM**, and **FAST** with the values **1**, **2**, and **3** to denote the fan speed.
- An **int** data field named **speed** that specifies the speed of the fan (the default is **SLOW**).
- A **boolean** data field named **on** that specifies whether the fan is on (the default is **false**).
- A **double** data field named **radius** that specifies the radius of the fan (the default is **5**).
- A String data field named **color** that specifies the color of the fan (the default is **blue**).
- A method **turnOn()** to turn on the fan.
- A method **turnOff()** to turn off the fan.
- A method **increaseSpeed()** that increases the speed of the fan unless the fan is running at highest speed. If it is running at highest speed then the method prints an error message "**Fan is already running at highest speed.**".
- A method **decreaseSpeed()** that decreases the speed of the fan unless the fan is running at lowest speed. If it is running at loweset speed then the method prints an error message "**Fan is already running at lowest speed.**".

. A method **changeSpeed(int speed)** that changes the speed of the fan to a new value using value of parameter **speed**. If the passed value is out of range then the method prints an error message "**Incorrect speed value.**".

. A method **changeColor(String color)** that changes the color of the fan to a new value using value of parameter **color**.

. A method **changeRadius(double radius)** that changes the radius of the fan to a new value using value of parameter **radius**.

. A method named **toString()** that returns a string description for the fan. If the fan is on, the method returns the fan speed, color, and radius in one combined string. If the fan is not on, the method returns the fan color and radius along with the string "fan is off" in one combined string.

Draw the UML diagram for the class and then implement the class. Write a test program that does the following:

- Creates two **Fan** objects.
- Assigns maximum speed, radius **10**, color **yellow** to the first object.
- Turns first object on.
- After first object is turned on, it increases its speed.
- Assigns medium speed, radius **5**, color **blue** to the second object.
- Turns second object on.
- Then decreases its speed twice,
- After that it turns it off.
- Displays the two objects by invoking their **toString()** method.

## Sample Run:

```
Fan is already running at highest speed.
Fan is already running at lowest speed.
-------------
Fan1:
speed 3
color yellow
radius 10.0
fan is on
-------------
Fan2:
speed 1
color blue
radius 5.0
fan is off
```

## Question 2

Design a class named **QuadraticEquation** for

a quadratic equation $ax^2 + bx + c = 0$. The class contains:

- Data fields **a**, **b**, and **c** (of type double) that represent three coefficients.
- A method named **calcDiscriminant()** that calculates and returns the discriminant (of type double), which is $b^2 - 4ac$ .
- The methods named **calcRoot1()** and **calcRoot2()** for calculating and returning two roots (of type double) of the equation

$$r_1 = \frac{-b + \sqrt{b^2-4ac}}{2a} \quad \text{and} \quad r_2 = \frac{-b - \sqrt{b^2-4ac}}{2a}$$

These methods are useful only if the discriminant is nonnegative. Let these methods return **0** if the discriminant is negative.

Draw the UML diagram for the class and then implement the class. Write a test program TestQuadraticEquation that prompts the user to enter values for *a*, *b*, and *c* and displays the result based on the discriminant. If the discriminant is positive, display the two roots. If the discriminant is 0, display one root. Otherwise, display "**The equation has no roots.**"
Use class names QuadraticEquation and TestQuadraticEquation

## Sample Run 1:

```
Enter a, b, c: 1 3 -4 ↵
The roots are 1.0 and -4.0
```

## Sample Run 2:

```
Enter a, b, c: 1 2 1 ↵
The root is -1.0
```

## Sample Run 3:

```
Enter a, b, c: 1 2 3 ↵
The equation has no roots
```