

## CSC260 Homework 6 (CH9)

- 50 points in total
  - 10 points from MyProgrammingLab questions (2 points per each question you select)
  - 20 points from class activities
  - 20 points from Programming questions.
- Submit your homework at Canvas/Assignments/HW6
- If you have any questions about the homework, please send me an email (start with CSC260 with a section number) or open a Discussion on Canvas.

### 3. Programming (20 points)

1. You should make the program that returns correct answer.
2. You should print out the results and paste them as a comment.
3. For getting outputs, use the inputs from the sample run. Some programs don't need an input, then print out and copy the results.
4. Copy only Java files for submission; copy only the Java files in the `programming` directory.
5. Students earn 100% when they get correct answers and copied results, 60% when they get wrong answers, 0% when they can't compile the Java source or no answers copied.

#### 3.1 QuadraticOOPTest.java (5 points)

Algebra: solve quadratic equations using OOP paradigm. This is a modification of the HW2 programming question.

The two roots of a quadratic equation  $ax^2 + bx + c = 0$  can be obtained using the following formula:

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

$b^2 - 4ac$  is called the discriminant of the quadratic equation. If it is positive, the equation has two real roots. If it is zero, the equation has one root. If it is negative, the equation has no real roots. Note that you can use `Math.pow(x, 0.5)` to compute  $\sqrt{x}$ .

You need to write the QuadraticOOP class to return the three solutions (5.0, -0.3819660112501051, and -2.618033988749895) from a equation  $x^2 + 3x + 1 = 0$ .

1. You need to write a constructor that has a, b, c (to represent an equation  $ax^2 + bx + c = 0$ ) as arguments.
2. You need to implmenet the getDiscriminant() method that returns the discriminant of the quadratic equation.
3. You need to implement getSolution1() that returns  $r_1$  and getSolution2() that returns  $r_2$ .
4. You don't have to implement any error processing routine.

### 3.2 FanTest.java (5 points)

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

- You need to make three constants named SLOW, MEDIUM, and FAST with values 1, 2, and 3 to denote fan speed.
- You need to have the following member fields (member variables or data fields)
  - A private int data field named speed that specifies the speed of the fan (the default is SLOW).
  - A private boolean data field named on that specifies whether the fan is on(the default is false).
  - A private 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).
- You need to have accessor methods to access all four member fields.
  - For example, the speed private data field should have getSpeed() and setSpeed().
- You need to make a default constructor (without any aruguments).

The main method in FanTest should returan all trues.

### 3.3 DateTest.java (10 points)

In this program, you learn to use Java's Date class to solve date related problems. Write a program that creates (1) a Date object, (2) sets its elapsed time to 10000, 100000, 1000000, 10000000, 100000000, 1000000000, 10000000000, and 100000000000 (read hint or textbook if you don't understand what does this mean) and (3) displays the date and time using the toString() method, respectively.

You should have the output as follows:

```
Wed Dec 31 19:00:10 EST 1969
Wed Dec 31 19:01:40 EST 1969
Wed Dec 31 19:16:40 EST 1969
Wed Dec 31 21:46:40 EST 1969
Thu Jan 01 22:46:40 EST 1970
Mon Jan 12 08:46:40 EST 1970
Sun Apr 26 13:46:40 EDT 1970
```

Sat Mar 03 04:46:40 EST 1973

**Hint:** You should instantiate a Date object, and invoke setTime() of the Date object to give argument 10000, 100000, and on.

```
// instantiate Date object
Date date = ???

int count = 1;
long time = 10000; // why it's important you use `long` data type?

while (count <= 8) {
    // use setTime(time) method of Date object to set time
    ???
    count++;
    time *= 10; // you can increase time to setup new time
    // print out
}
```