Lab 3

CST 338

Task 1: CourseInfoDemo

Download CourseInfo.java and CourseInfoDemo.java (both files are linked). Create a Java project in *IntelliJ IDEA* with the two files. Then, change the `readInfo()` method to read the scores of all students enrolled and determine the average. The `printInfo()` method should display the average of the scores of all students enrolled. The following presents a sample execution of the program. You may need to change the `CourseInfoDemo.java` file appropriately to run your program as below (bold text implies input):

```
Enter course title: Software Design
Enter instructor name: Avner
Enter enrollment: 3
Enter a score:
100
Enter a score:
98
Enter a score:
97

Course title: Software Design
Instructor: Avner
Enrollment: 3
Average: 98.33
```

Task 2: SoftHouse

Within a new project called SoftHouse create a Java program called BillingDialog.java as below. (File is linked.)

```
public class BillingDialog {
  public static void main(String[] args) {
    System.out.println("Welcome to CSUMB software house!");
    Bill yourBill = new Bill();
    yourBill.inputTimeWorked();
    yourBill.updateFee();
    yourBill.outputBill();
    System.out.println("Thanks for doing business with us.");
  }
}
```

The class `BillingDialog` uses a user-defined class `Bill`. To define the class, you have to create `Bill.java`.

(1) Compile and run `BillingDialog.java` with the input of 2 hours and 45 minutes. Your execution result would be like below (bold text implies input):

```
Welcome to CSUMB software house!
Enter number of full hours worked
followed by number of minutes:
2 45
Time worked:
2 hours and 45 minutes
Rate: $150.0 per quarter hour.
Amount due: $1650.0
Thanks for doing business with us.
```

(2) Modify the `Bill` class so that the class can support a discount coupon number. If a user entered a coupon number such as 1234, your program should provide 10% discount of the fee. All other coupon numbers should be ignored. The following presents a sample run:

```
Welcome to CSUMB software house!
Enter number of full hours worked
followed by number of minutes:
2 45
Time worked:
2 hours and 45 minutes
Enter a coupon number:
1234
Your coupon is valid. (10% discount)
Rate: $135.00 per quarter hour.
Amount due: $1485.00
Thanks for doing business with us.
```

Task 3: GameCar

Create a Java project called `GameCar` and create a Java program named `GameCar.java` to define a class. When you define the class, you have to provide the following methods so that a user of the class can invoke these methods.

```
/*  This method reads a game car's name, color,
    and current position from a user. */
void getCarInfo()

/*  This method makes the car move forward by
    adding the distance to the current position. */
void moveForward(int distance)

/*  This method makes the car move backward
    by subtracting the distance from the current position. */
void moveBackward(int distance)

/*  This method compares the position of the
    car with another car's position and prints the winner. */
void race(GameCar anotherCar)
```

In addition to the above methods, you need to add more methods to run the GameCarDemo.java program properly. This is part of the lab assignment.
The following class named `GameCarDemo` presents sample usage of the `GameCar` class. If you find any problems in the class, please fix them.

```
public class GameCarDemo {
  public static void main(String[] args) {
    GameCar car1 = new GameCar("MyBuddy", "Red", 19);
    GameCar car2 = new GameCar();
    car1.moveForward(5);
    car1.moveBackward(2);
    car1.setColor("Black");
    System.out.println(car1);
    car2.getCarInfo();
    car2.moveForward(7);
    car2.moveForward(3);
    System.out.println(car2);
    car1.race(car2);
    car1.moveForward(10);
    car1.race(car2);
    car1.setCar("BlueJay", "Blue", 30);
    car1.race(car2);
    System.out.println("Same cars? " + car1.equals(car2));
  }
}
```

This is a sample execution of the class `GameCarDemo`. From this script, infer the appropriate output messages for the corresponding methods.

```
Car Information: MyBuddy (Black), Current Position: 22
Enter the car name: BlueJay
Enter the color: Blue
Enter the starting position: 20
Car Information: BlueJay (Blue), Current Position: 30
BlueJay Win: MyBuddy (22) vs. BlueJay (30)
MyBuddy Win: MyBuddy (32) vs. BlueJay (30)
Tie: BlueJay (30) vs. BlueJay (30)
Same cars? true
```

Based on the information of the description and execution result, implement the class.
When you finish your lab, submit the Java source code on iLearn.