2021F CS234 Computer Science II
Lab 6
Total points: 100

1) P8.9 (50 points)

Write a ***class definition*** named **Bug** (i.e., .java file **without the main method**) that models a bug moving along a horizontal line. The bug moves either to the right or left. **Initially**, the bug moves to the **right** starting at position **10**, but it can turn to change its direction. In each move, its position changes by one unit in the current direction.
Therefore, you need to have **one instance** variable to keep track of the current position and **one instance variable** to keep track of the current direction (right or left). **No extra instance variables are allowed.**

Provide the following methods: 1) a method to **move** the bug, 2) a method to **turn** direction, and 3) a method to **get** the **position** of the bug. These methods are the **only allowed** to modify the instance variables (i.e., no direct access to these instance variables).
No extra methods are allowed.

I will **use** a separate **tester program** (separate .java file with main() method) to test your class definition.
The main method for **my** tester program is the following:

```
public static void main(String[] args) {

    Bug myBug = new Bug();
    myBug.move();
    myBug.move();
    myBug.turn();
    myBug.move();
    myBug.turn();
    myBug.move();
    myBug.turn();
    myBug.move();
    myBug.move();
    System.out.println("Expected position: 10");
    System.out.println("Current position: " + myBug.getPosition());
}
```

The output of my tester program is:

```
Expected position: 10
Current position: 10
```

You can create your own tester program to test your class definition.
From my tester program you **can take the names for your methods.**
Be careful, I will use this same tester program to test your class definition therefore the **names** for the class and methods should be **the same** in your class definition.

2) Based on P8.7 (50 points) Implement a **class definition** named **Student** (i.e., a .java file ***without the main method***). For this exercise, a student has a **name** and a **list** of **scores.**
You need to model this information using **one instance variable** for the name, and **one instance variable** for the list of scores. **No extra instance variables are allowed.**

You need to define methods to set 1) the name, 2) get the name, 3) add a score, 4) get the total of scores (the addition of all scores), 5) get the average score, and 5) get the higher score. These methods are the **only allowed** to modify the instance variables (i.e., no direct access to these instance variables). **No extra methods are allowed.**

I will **use** a separate **tester program** (separate .java file with main() method) to test your class definition.
The main method for **my** tester program is the following:

```
public static void main(String[] args) {

    Student john = new Student();
    Student emma = new Student();

    john.setName("John");
    emma.setName("Emma");

    emma.addScore(8.5);
    emma.addScore(9.5);
    System.out.printf("Student: %s\n", emma.getName());
    System.out.printf("Total score: %.2f\n", emma.getTotalScore());
    System.out.printf("Average score: %.2f\n", emma.getAverageScore());
    System.out.printf("Higher score: %.2f\n", emma.getHighScore());

    john.addScore(9.3);
    john.addScore(7.4);
    john.addScore(8.5);
    john.addScore(6.6);
    System.out.printf("Student: %s\n", john.getName());
    System.out.printf("Total score: %.2f\n", john.getTotalScore());
    System.out.printf("Average score: %.2f\n", john.getAverageScore());
    System.out.printf("Higher score: %.2f\n", john.getHighScore());
}
```

The output of my tester program is:

```
Student: Emma
Total score: 18.00
Average score: 9.00
Higher score: 9.50
Student: John
Total score: 31.80
Average score: 7.95
Higher score: 9.30
```

You can create your own tester program to test your class definition.
From my tester program you **can take the names for your methods.**
Be careful, I will use this same tester program to test your class definition therefore the **names** for the class and methods should be the same in your class definition.

Submission details:

Upload a **single ZIP** file.
Name your file as follows: **Lab6_Lastname_Firstname.zip**
There is a **10% points deduction** if your file does not have the correct name.

Your **.zip** file must contain the following:

1. Your **.java** source files for your **class definitions** (.java files without the **main** method. No .class files). **Do not send the tester program.**
2. A **SINGLE PDF** with screenshots from your programs running (10% points deduction if you don't submit a **SINGLE** PDF file). Do not send .jpg files.

For this lab, you do **not need to submit the .txt** file with your instructions. **Why?** Because I will use my tester programs to use your classes. Therefore, it is **extremely important** that your **class and method names** are **correct.**

In each .java file, **write** as a multiline comment at the beginning of the file the following:

1. Your name
2. The course section

The **zip** file must be uploaded to Canvas. I do not accept answers via email.
I do not accept image files; it must be a PDF file.

Make sure to check the **due date** for this activity on Canvas. Try to submit it before the due date so you can have time to check for improvements.
Make sure you are **submitting the correct files**. I will grade the files uploaded to Canvas.
Make sure you **test** your **classes** with a similar **tester program** as the ones I am showing in this lab (i.e., a .java program with a main method where you create objects from your class).
Use the *javac* and *java* commands **before** submitting your solution. You **don't submit** your **tester** program.
Make sure to review the **grading rubric**.
Late submissions are not allowed.

If you have questions, contact me before making assumptions about what you need to do for solving this assignment.
If you have questions, contact me before going with a tutor. I need to know what issues you are having to adjust the content of my lectures.