CSCI 3400
Fall 2023
Lab 01: Brushing up on Java
Assigned date: 08/31/2023 (Thursday)
Due date: 09/07/2023 (next Thursday, 11:59 PM, EST)
Total points: 60

Learning Goal: In this lab, we practice designing a Comparable Class and handling basic file operations.

You have **three files**:
`Robot.java`, `TestMain.java`, and `robotList.txt`

Create a package, 'lab01', and add the source codes there. Run it. Now follow the instructions given below.

`Robot Class:`

1. Given the `Robot` class, add necessary getters and setters (accessors and mutators) for all the existing properties. If you have your own version of the Robot class/interface, feel free to use it. **(10 points)**

2. Add the following methods: **(10 points)**

   - `public void alterFaceMask()` to alter the facemask status of the robot (if the mask is on, take it off and vice versa).

   - `public String toString()` that returns a statement like the following:
     lieutenantRobot is a rank 1 robot.

     Here lieutenantRobot is the name, and 1 is the rank of the corresponding robot object.

`robotList.txt:`

This file contains several lines. Each line contains information about a new robot. For now, assume that each line has all the fields necessary to create a robot object in the following sequence:

`Name Rank Salary FaceMaskStatus SuitColor`

`TestMain Class:`

a) In the **main(…)** method, add necessary instructions to read from the robotList.txt file. After reading each line, create a new robot object and add it to a list. Ensure that your code can read the input file stored in any location (I mean, the user needs to specify the exact location, a.k.a. path to the file). **(10 points)**

Hint: You may use ArrayList like the following:

```
 ArrayList<Robot> roboList = new ArrayList<Robot>();
```

b) Sort the list of robots based on their ranks (lowest to highest). **(10 points)**

(Hint0: Collections class has a sort method. To be able to use that, we have to make the Robot class comparable.)

(Hint1: If you are not familiar with the Comparable interface, check this: https://www.javatpoint.com/Comparable-interface-in-collection-framework)

c) Print the robots in your list. **(10 points)**

d) Sort by another property.

If you are familiar with comparators, create a Comparator object that compares two Robots by their names (alphabetic order). Now use this comparator to sort the robot list by names and print them. **(10 points)**

Hint: There are several ways of doing this. One possible way may look like the following:

```
// Add this class in the same (default) package.

class CompareBySalary implements Comparator<Robot>{

      @Override
      public int compare(Robot i, Robot j) {

             if(i.getSalary() > j.getSalary())
                     return 1;
             else if (i.getSalary() < j.getSalary())
                     return -1;
             else
                     return 0;
      }

//Now, in the body of the main, add these lines:

Collections.sort(roboList, new CompareBySalary());

System.out.println("Here is the list of robots sorted by salary: ");
```

```
for(Robot r: roboList) {

             System.out.println(r);

             }
```

Submission:

Add your name after your instructor's name and add citations (any resource that you have used while writing code, for example any website that you have looked for).

Submit the source codes (TestMain.java and Robot.java, and/or any other necessary file).