

## COMP2120 - Fall 2020

### Lab 4 Activities

#### Activity 1.

Implement a class `Robot` that simulates a robot wandering on an infinite plane. The robot is located at a point with integer coordinates and faces north, south, east, or west. Supply methods:

```
public void turnLeft()
public void turnRight()
public void move()
public Point getLocation()
public String getDirection()
```

The `turnLeft` and `turnRight` methods change the direction but not the location. The `move` method moves the robot by one unit in the direction it is facing. The `getDirection` method returns a string "N", "S", "E", "W". For the new object, the initial location and facing should be specified. For the initial facing, 0 means north, 1 means east, 2 means south, and 3 means west.

A tester class, `RobotTester`, has been provided as follows to help you how to implement the `Robot` class. Therefore, implement the `Robot` class such that the following `RobotTester` class can be executed without any changes.

```
import java.awt.Point;

/**
 * A class to test the Robot class.
 */
public class RobotTester
{
    /**
     * Tests the methods of the Robot class.
     * @param args not used
     */
    public static void main(String[] args)
    {
        // Create a new Robot object with the initial position of (5,5) and
        // initial direction to the East (1).
        Robot robot = new Robot(new Point(5, 5), 1);

        robot.move(); // 6, 5, 1
        robot.turnRight(); // 6, 5, 2
        robot.move(); // 6, 6, 2
        robot.move(); // 6, 7, 2
        robot.turnRight(); // 6, 7, 3
        robot.move(); // 5, 7, 3
        robot.move(); // 4, 7, 3
        robot.turnLeft(); // 4, 7, 2
        robot.move(); // 4, 8, 2
    }
}
```

```

        Point location = robot.getLocation();
        System.out.println("Location: " + location.x + ", " + location.y);
        System.out.println("Expected: 4, 8");
        System.out.println("Direction: " + robot.getDirection());
        System.out.println("Expected: S");
    }
}

```

## Activity 2.

A car dealership wants to have a system to keep the records of all its cars. We already had some experience creating a class for *Car*. Use what you learned to create a class *Car*, with some required information, such as make, model, year, transmission, seats, maximum speed, wheels, type (sedan, SUV, Crossover, ...), etc. This dealership needs to keep the information of all its cars. Therefore, we should create a class *Dealership*, with some profile information such as name, location, managers, employees, etc. It also should be able to keep the information of all its cars.

Your tasks are creating a class for *Car*, a class for *Dealership*, and also a tester class to be able to create one or more dealerships, each of which have one or more cars. Note that the number of managers, employees, and cars of a dealership are not known in advance and therefore you should consider this fact in your class development.