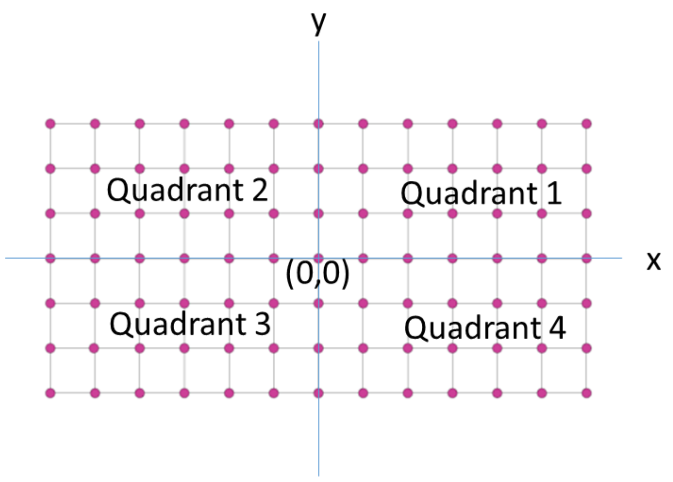
CS 3300 Assignment 2
Spring 2020
Due Date: 11:59 PM of 02/12/2020

What to submit:

- You will submit three .java files as described below. Do not submit the entire project folder or .class files.
- Also, submit screenshots of the output of your code.

Codes will be compared using document-comparing tools; any similarities between codes will lead to an "F" to the involving parties.

In this assignment, you will simulate the motion of a random robot and a "biased" robot moving in an infinite two-dimensional grid (shown below). In each time step, the robot has four options: move one-step (each step is of length 1) in one of the four directions (East, West, North, and South). Diagonal motion is not permitted. The grid has four quadrants (1, 2, 3, and 4) as shown.



Your goal is to create two JAVA classes (one java class = one .java file). `RandomRobot` and `BiasedRobot`. Also, create a third class **`RobotTest`** for testing purposes. The descriptions are below.

`RandomRobot` class:

- Member variables:
  - Position coordinates x and y as integers.
  - Current direction of motion as a string ("East", "West", "North", or "South").
  - Current quadrant (1, 2, 3, 4), the robot is located as an integer. If the robot is on x or y-axis, quadrant should be assigned 0.
- Member functions

- Constructor `RandomRobot():` creates a robot located in the origin (0,0) and random direction of movement.
- Constructor `RandomRobot(int x, int y):` creates a robot with location as given in the constructor arguments. The current direction is randomly determined.
- `move()`
  - Simulates the movement of the robot for one time step by randomly moving the robot in one of the four possible directions.
  - Update all the member variables after moving.
- `simulate(int numSteps)`
  - Simulates the movement of the robot for the given number of time steps `numSteps`. Use `move()` function.
- Accessor and mutator functions of all appropriate member variables should be included. Note: the quadrant depends on (x,y). If you update (x,y), quadrant should be updated.
- **Add member functions as needed.**

`BiasedRobot` class:

Similar to the `RandomRobot` class except that the `move()` function is different.

- In the constructors, the current direction of the biased robot is chosen randomly.
- `move()`
  - Simulates the movement of the biased robot for one time step by moving the robot in one of the four possible directions with the following probability:
    - “East”: 50%
    - The remaining 50% probability should be equally distributed across the three remaining directions.
  - Update all the member variables after moving.
- **Add member functions as needed.**

`RobotTest` Class:

This class will only have the main function. Inside the main function, the following should be done.

1. Create an object of `RandomRobot` class using the first constructor `RandomRobot()`. Simulate its movement for 500 time steps.
2. Repeat step #1 1000 times. Out of 1000 simulations, find out how many times each quadrant the robot’s final location belongs.

3. Create an object of `BiasedRobot` class using the first constructor `BiasedRobot()`. Simulate its movement for 500 time steps.
4. Repeat step #3 1000 times. Out of 1000 simulations, find out how many times each quadrant the biased robot’s final location belongs.