

# ROBOT SIMULATOR

## 1. DESCRIPTION

The application is a simulation of a toy robot moving on a square tabletop, of dimensions 5 units x 5 units. There are no other obstructions on the table surface. The robot is free to roam around the surface of the table, but must be prevented from falling to destruction. Any movement that would result in the robot falling from the table must be prevented, however further valid movement commands must still be allowed.

## 2. ASSIGNMENT

Create an application that can read commands of the following form:

PLACE X,Y,F	Put the robot on the table in position X,Y and facing NORTH, SOUTH, EAST or WEST
MOVE	Move the robot one unit forward in the direction it is currently facing
LEFT	Rotate the robot 90 degrees left
RIGHT	Rotate the robot 90 degrees right
REPORT	Announce the X,Y and F of the robot. This can be in any form, but standard output is sufficient

- The origin (0,0) can be considered to be the SOUTH WEST most corner.
- The first valid command to the robot is a PLACE command, after that, any sequence of commands may be issued, in any order, including another PLACE command. The application should discard all commands in the sequence until a valid PLACE command has been executed.
- A robot that is not on the table can choose to ignore the MOVE, LEFT, RIGHT and REPORT commands.
- Input should be in the form of a text file.
- Provide test data to exercise the application.
- The solution should be sufficiently unit tested. Include unit tests.
- It is not required to provide any graphical output showing the movement of the toy robot.

## 3. CONSTRAINTS

- The toy robot must not fall off the table during movement. This also includes the initial placement of the toy robot. Any move that would cause the robot to fall must be ignored.

## 4. DELIVERABLES

- The source files, without any dependencies except for test dependencies.
- The test data and any test code.

## 5. EXAMPLES

a)

PLACE 0,0,NORTH

MOVE

REPORT

Output: 0,1,NORTH



b)  
PLACE 0,0,NORTH  
LEFT  
REPORT  
Output: 0,0,WEST

c)  
PLACE 1,2,EAST  
MOVE  
MOVE  
LEFT  
MOVE  
REPORT  
Output: 3,3,NORTH

**GOOD LUCK!**

