|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **LEARNING PROFILE FOR ASSIGNMENT#\_2\_\_\_\_\_ AND QUESTION#\_9\_\_\_\_\_\_\_** | | | | | |
| *Name* | *:* | *Steven Morrissey* | *Due Date* | *:* |  |
| *Student ID* | *:* | *3300222* | *Submission Date* | *:* |  |

**1. Problem Statement:**

Create a 10x10 matrix as a 2D array. Assume that a robot is placed in position [0, 0]. Now randomly generate a move. The move could take the robot to one of the eight possible adjacent slots – {up, down, left, right, left-upcorner, left-down-corner, right-up-corner, and right-down-corner} – these slots are represented by {1, 2, 3, 4, 5, 6, 7, 8}. However, at [0, 0], the robot only has three possible slots to move to – right, down, right-down-corner. Create another robot called R2 and place it on [9, 9]. Now randomly generate an integer in the range of [1 to 8]. This first random integer corresponds to a possible move for Robot R1. If the move is valid, then move R1 to its new slot. A move is invalid if it takes the robot out of bounds of the [10x10] matrix. If the move is invalid, then keep generating random integers until a valid move is found. Repeat this procedure for the second Robot R2. If both R1 and R2 are in the same slot, then stop, print the final slot, print the sequence of random numbers that led R1 to this slot, and the print the sequence of random numbers that led R2 to the same slot. Implement this program with a Robot class and a MovingRobot subclass.

**2. Description of the Code:**

**Most of the logic is done in the play method(). The players use the random method to “roll the dice” and the convert method to translate between dice roll and literal name of the outcome. I also included player 1 & 2 in the convert method since for this exercise we let the program generate the players and in the api there are two final static players defined.**

**3. Errors and Warnings:**

Table 1: List of Errors and Warnings Encountered in the Program

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Errors / Warnings** | **Details** | **How I solved them** |
| 1 | none |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

**4. Sample Input and Output:**

**5. Discussion:**