

Homework 7

Reading

Read Chapter 10 in **Introduction to Computing using Python: An Application Development Focus, Second Edition** by Ljubomir Perković.

Logistics

If you want to do the assignment on your own computer, you will need to download and install Python. There is a link for downloading Python under Content -> Course Information on D2L.

In this class programming assignments may be completed in consultation with up to two other classmates. You must identify the classmates with whom you collaborate in the comment box when you submit on D2L. You must also list their names at the top of the assignment. The total number of collaborators on any assignment **may not exceed two other people**. Please see the Collaboration Guidelines document found under Content -> Course Information on D2L for more details on what is or is not allowed when working on homework.

Remember that **everyone** submitting the assignment must be able to explain **all** of the code submitted, regardless of the type of collaboration that occurred. Anyone submitting code they cannot explain is violating the Academic Integrity policy and will earn a 0 on the assignment.

Assignment

To begin the assignment, download the **HW7.py** file and the **HW7Test.py** found on the D2L site. Note that the **HW7.py** contains just the first line of a class definition for a robot class and a pass line. You will need to remove the pass and add definitions for all the functions described below. A robot is located on the (x,y) plane and faces one of four directions (up, down, left, or right). You must fill in all the missing methods described below.

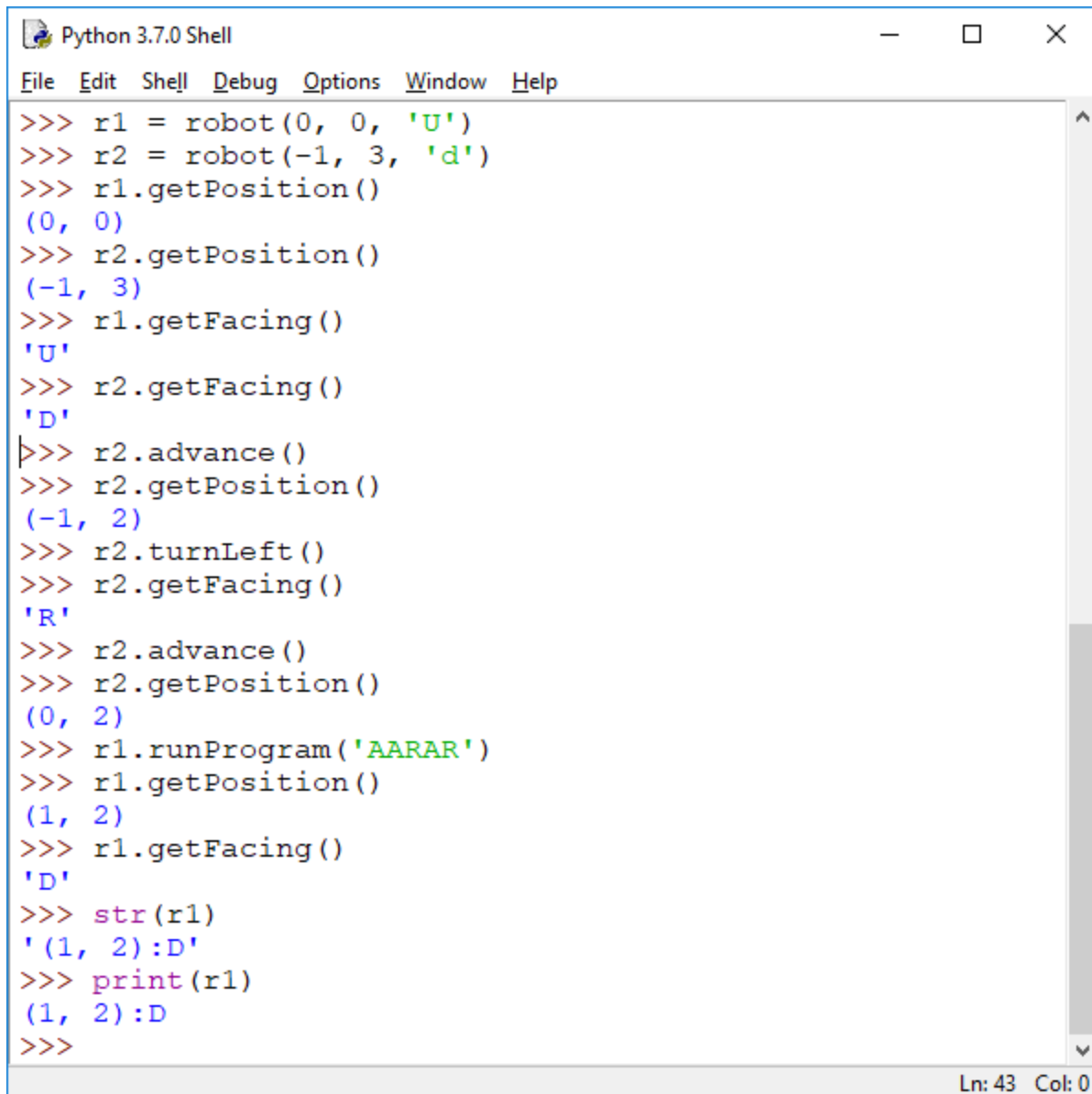
1. The constructor takes 3 arguments. The first two are integers and correspond to the initial x and y coordinates of the robot. The third argument is a single character string containing one of the characters U, u, D, d, R, r, L, or l. These characters correspond to the initial direction that the robot is facing (up, down, right, or left).
2. The robot class has a turnLeft method that takes no arguments. The result of calling this method on a robot is that it rotates ¼ turn in the clockwise direction. For example, if the robot is currently facing left, then after calling turnLeft it will be facing down. The method does not return anything.
3. The robot class has a turnRight method that takes no arguments. The result of calling this method on a robot is that it rotates ¼ turn in the counter clockwise direction. For

example, if the robot is currently facing left, then after a call to `turnRight` it will be facing up. The method does not return anything.

4. The robot class has an `advance` method that takes no arguments. The result of calling this method on a robot is that it moves one unit in the direction it is facing. For example, if the robot is currently at (10, 20) and is facing down, then after calling `advance` the robot will be at (10, 19). The method does not return anything.
5. The robot class has a `getPosition` method that takes no arguments. It returns a tuple containing two integers, the x and y position of the robot.
6. The robot class has a `getFacing` method that takes no arguments and returns one of 4 strings: 'U', 'D', 'L', or 'R', indicating which direction the robot is facing. Note: while the constructor accepts lower case letters, `getFacing` always returns upper case letters.
7. The robot class has a `runProgram` method that takes a string consisting of the characters A, R, or L. These letters correspond to issuing the commands to advance (`advance` method), to turn to the right (`turnRight` method) or to turn to the left (`turnLeft` method). The result of calling the `runProgram` method with such a string is the same as if the corresponding methods were called on the robot in the order they appear in the string. So a robot given the program 'ARAL' should make the following sequence of moves: `advance()`, `turnRight()`, `advance()`, and `turnLeft()`.
8. The `str()` constructor when called on a robot should result in a string with the format '(x, y):d' where x is the current x position, y is the current y position, and d is the current direction. For example, if the robot r's position is (-4, 3) and it is currently facing left, then `str(r)` should return the string '(-4, 3):L'.

Sample Execution

Below is an example of how a robot behaves interactively in the interpreter.



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
>>> r1 = robot(0, 0, 'U')
>>> r2 = robot(-1, 3, 'd')
>>> r1.getPosition()
(0, 0)
>>> r2.getPosition()
(-1, 3)
>>> r1.getFacing()
'U'
>>> r2.getFacing()
'D'
>>> r2.advance()
>>> r2.getPosition()
(-1, 2)
>>> r2.turnLeft()
>>> r2.getFacing()
'R'
>>> r2.advance()
>>> r2.getPosition()
(0, 2)
>>> r1.runProgram('AARAR')
>>> r1.getPosition()
(1, 2)
>>> r1.getFacing()
'D'
>>> str(r1)
'(1, 2):D'
>>> print(r1)
(1, 2):D
>>>
```

Ln: 43 Col: 0

Testing

Included in the dropbox is a file called HW7Tests. You are required to run that test file on your HW7.py file, and submit a screenshot of the result. Remember that to run HW7Test.py it must be in the same folder as HW7.py. Open HW7Test.py in python and then hit F5 or select Run Module from the Run menu.

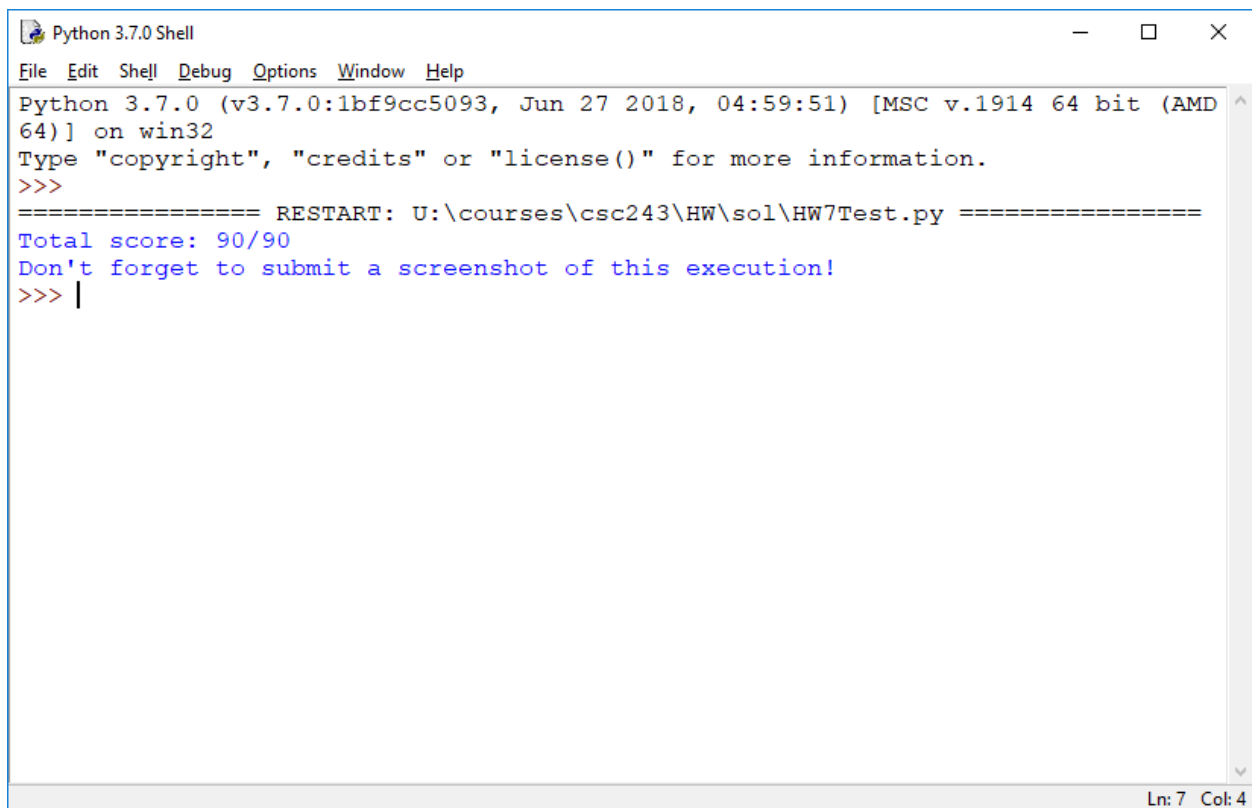
There are 9 tests total and each one is worth 10 points. Note the test file only reports pass or fail. If the test fails, you will need to add some extra print statements to track down where the test is failing. The final 10 points are for the screenshot of you running the test.

Submitting the assignment

You must submit the assignment using the HW7 folder of the dropbox on D2L. Submit your HW7.py file as well as a screenshot of running HW7Test.py. Failure to submit the screenshot will result in a 10 point deduction to your score. Submissions after the deadline will be automatically rejected by the system and will receive a grade of 0.

Grading

Each of the 9 tests is worth 10 points for a total of 90 points. The screenshot of you running the test file (regardless of whether or not the tests pass) is worth 10 points. So submit the screenshot! Below is a screenshot of what you should see on a correct execution of HW7Test:



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD
64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: U:\courses\csc243\HW\sol\HW7Test.py =====
Total score: 90/90
Don't forget to submit a screenshot of this execution!
>>> |
```

Ln: 7 Col: 4