

[Courseware \(/courses/MITx/6.00x/2012_Fall/courseware/\)](/courses/MITx/6.00x/2012_Fall/courseware/)[Course Info \(/courses/MITx/6.00x/2012_Fall/info/\)](/courses/MITx/6.00x/2012_Fall/info/)[Textbook \(/courses/MITx/6.00x/2012_Fall/book/0/\)](/courses/MITx/6.00x/2012_Fall/book/0/)[Discussion \(/courses/MITx/6.00x/2012_Fall/discussion/forum/\)](/courses/MITx/6.00x/2012_Fall/discussion/forum/)[Wiki \(/courses/MITx/6.00x/2012_Fall/course_wiki/\)](/courses/MITx/6.00x/2012_Fall/course_wiki/)[Progress \(/courses/MITx/6.00x/2012_Fall/progress/\)](/courses/MITx/6.00x/2012_Fall/progress/)

PROBLEM 2: THE STANDARDROBOT CLASS : 15.0 POINTS

Each robot must also have some code that tells it how to move about a room, which will go in a method called

`updatePositionAndClean`.

Ordinarily we would consider putting all the robot's methods in a single class. However, later in this problem set we'll consider robots with alternate movement strategies, to be implemented as different classes with the same interface. These classes will have a different implementation of `updatePositionAndClean` but are for the most part the same as the original robots. Therefore, we'd like to use inheritance to reduce the amount of duplicated code.

We have already refactored the robot code for you into two classes: the `Robot` class you completed in Problem 1 (which contains general robot code), and a `StandardRobot` class that inherits from it (which contains its own movement strategy).

Complete the `updatePositionAndClean` method of `StandardRobot` to simulate the motion of the robot after a single time-step (as described on the Simulation Overview page).

```
class StandardRobot(Robot):
    """
    A StandardRobot is a Robot with the standard movement strategy.

    At each time-step, a StandardRobot attempts to move in its current direction; when
    it hits a wall, it chooses a new direction randomly.
    """
    def updatePositionAndClean(self):
        """
        Simulate the passage of a single time-step.

        Move the robot to a new position and mark the tile it is on as having
        been cleaned.
        """
```

Before moving on to Problem 3, check that your implementation of Standard Robot works by uncommenting the following line under your implementation of StandardRobot. If you're running Python 2.6 (note that Enthought runs Python 2.7), you will need to change the import lines at the top of `ps7.py`. Make sure that as your robot moves around the room, the tiles it traverses switch colors from gray to white.

```
testRobotMovement(StandardRobot, RectangularRoom)
```

When you've checked that your robot moves correctly, make sure to comment out the above line.

GRADER IS CURRENTLY DOWN; IT WILL BE AVAILABLE SOON!

In the meantime, you can work on implementing this problem on your own machine.

CheckSave



[Find Courses \(/courses\)](/courses) [About \(/about\)](/about) [Blog \(http://blog.edx.org/\)](http://blog.edx.org/) [Jobs \(/jobs\)](/jobs) [Contact \(/contact\)](/contact)



<http://youtube.com/user/edxonline>



<https://plus.google.com/108235383044095082735>



<http://www.facebook.com/EdxOnline>



<https://twitter.com/edXOnline>

[terms of service \(/tos\)](/tos) [privacy policy \(/privacy\)](/privacy) [honor code \(/honor\)](/honor) [help \(/help\)](/help)