# 20.12. Testing classes

**Note**

This page depends on the use of the test module, which is introduced in the testing chapter. If you haven't covered that chapter yet, you will want to delay reading this page until you do.

To test a user-defined class, you will create test cases that check whether instances are created properly, and you will create test cases for each of the methods as functions, by invoking them on particular instances and seeing whether they produce the correct return values and side effects, especially side effects that change data stored in the instance variables. To illustrate, we will use the Point class that was used in the introduction to classes.

To test whether the class constructor (the `__init__`) method is working correctly, create an instance and then make tests to see whether its instance variables are set correctly. Note that this is a side effect test: the constructor method's job is to set instance variables, which is a side effect. Its return value doesn't matter.

A method like `distanceFromOrigin` in the `Point` class you saw does its work by computing a return value, so it needs to be tested with a return value test. A method like `move` in the `Turtle` class does its work by changing the contents of a mutable object (the point instance has its instance variable changed) so it needs to be tested with a side effect test.

Try adding some more tests in the code below, once you understand what's there.

Save & Run Original - 1 of 1 Show in CodeLens

```
class Point:
    """ Point class for representing and manipulating x,y coordinates. """

    def __init__(self, initX, initY):

        self.x = initX
        self.y = initY

    def distanceFromOrigin(self):
        return ((self.x ** 2) + (self.y ** 2)) ** 0.5

    def move(self, dx, dy):
        self.x = self.x + dx
        self.y = self.y + dy

```

Activity: 1 -- ActiveCode (ac19_3_1)

**Check your understanding**

test-3-1: For each function, you should create exactly one test case.

○ A. True

● B. False

Check me Compare me

✓ It's a good idea to check some extreme cases, as well as the typical cases.

Activity: 2 -- Multiple Choice (question19_3_1)

test-3-2: To test a method that changes the value of an instance variable, which kind of test case should you write?

○ A. return value test

● B. side effect test

Check me Compare me

✓ The move method of the Point class above is a good example.

Activity: 3 -- Multiple Choice (question19_3_2)

test-3-3: To test the function maxabs, which kind of test case should you write?

```
def maxabs(L):
    """L should be a list of numbers (ints or floats). The return value should be the maximum absolute value of the numbers in L."""
    return max(L, key=abs)
```

● A. return value test

○ B. side effect test

Check me Compare me

✓ You want to check if maxabs returns the correct value for some input.

Activity: 4 -- Multiple Choice (question19_3_3)

test-3-4: We have usually used the `sorted` function, which takes a list as input and returns a new list containing the same items, possibly in a different order. There is also a method called `sort` for lists (e.g. `[1,6,2,4].sort()` ). It changes the order of the items in the list itself, and it returns the value `None` . Which kind of test case would you use on the sort method?

○ A. return value test

● B. side effect test

Check me    Compare me

✓ You want to check whether it has the correct side effect, whether it correctly mutates the list.

Activity: 5 -- Multiple Choice (question19_3_4)

You have attempted 6 of 5 activities on this page

20.11. Thinking About Classes and Instances">

20.13. A Tamagotchi Game">

20.11. Thinking About Classes and Instances">

✓ Completed. Well Done!

20.13. A Tamagotchi Game">Next Section - 20.13. A Tamagotchi Game

© Copyright 2017 bradleymiller. Created using Runestone 4.1.17.

| Back to top