

```
import unittest

class TestClass(unittest.TestCase):
    def test_classes(self):
        # Create a class called ``Coyote``
        # Accept a name in the constructor
        # Create a coyote object called wilee with the name 'Wile E. Coyote'
        # =====
        class Coyote(object):
            def __init__(self, name):
                self.name=name
        wilee = Coyote('Wile E. Coyote')
        self.assertEqual(isinstance(wilee, Coyote))

        # Add a method ``speak`` to a Coyote class that
        # accepts a string and returns:
        # '${name} said, "${string}"
        # where "${string}" is an argument passed into the method
        # =====
        class Coyote(object):
            def __init__(self, name):
                self.name=name
            def speak(self, message):
                return self.name + ' said, "' + message + '"'
        wilee = Coyote('Wile E. Coyote')
        self.assertEqual(wilee.speak('Being a genius certainly has its
advantages. '),
                        'Wile E. Coyote said, "Being a genius certainly has its
advantages. "')

        # Subclass - create a subclass ``RoadRunner``
        # of ``Coyote`` that has the same constructor
        # but ``beep`` returns:
        # '${name} said, "${string}"
        # =====
        class RoadRunner(Coyote):
            def beep(self, beeps):
                return self.name + ' said, "' + beeps + '"'
        rr = RoadRunner("Road Runner")
        self.assertEqual(rr.beep('Beep. Beep. '), 'Road Runner said, "Beep. Beep. "')

if __name__ == '__main__':
    unittest.main()
```