BEGINNER'S DESIGN RECIPE

1. Read the problem. What kinds of data does the function consume? What kind of data does it produce? Write a short purpose statement. List all required input parameters (use meaningful names). Provide data kinds for each parameter, including the return value.

Write a function that computes the area of a rectangle, given the width and height.

```
        """
        computes the area of the given rectangle as width times
        height   [PURPOSE]

        width -- number        [CONTRACT]
        height -- number


        return -- number
        """
```

2. Write the function header, using the information from the contract.

```
    def area(width, height):
        ...
```

3. Make up examples of inputs. What should your program produce for those inputs? Write these as unit tests using `assertEqual`.

```
class MyTest(unittest.TestCase):
    # preparing to test
    def setUp(self):
        print("test setup")
    # ending the test
    def tearDown(self):
        print("test teardown")

    def test(self):
        self.assertEqual(area(2,3), 6)
        self.assertEqual(area(2,4), 8)
```

4. Now write your function body. Look closely at your examples, and try to see the pattern. How can the inputs be turned into the desired outputs? If your purpose statement indicates that different outputs should occur in different conditions,

consider using an if/elif/else structure [Chapter 4]. Consider what the condition should be for each case, and what the result should be, separately.

```
def area(width, height):
    ...
    return width * height
```

5. Now put it all together into one Python file, see the next page for *example*.

```
import unittest

def area(width,height):
    """
    Computes the area of the given rectangele
    as width times height [PURPOSE]

    width -- number [CONTRACT]
    height -- number

    return -- number
    """
    return width*height

class MyTest(unittest.TestCase):
    # preparing to test
    def setUp(self):
        print("test setup")
    # ending the test
    def tearDown(self):
        print("test teardown")

    def test(self):
        self.assertEqual(area(2,3), 6)
        self.assertEqual(area(2,4), 8)

unittest.main()
```