

```
from unittest import TestCase, main
```

```
class Test_get_input(TestCase):  
    def test_get_file_paths(self):  
        self.assertTrue("get_file_paths_" in globals() )  
        self.assertEqual(len(get_file_paths_()), 2)  
        self.assertTrue(path.isfile(get_file_paths_()[0]))  
        self.assertTrue(path.isfile(get_file_paths_()[1]))  
    def test_get_sheet(self):  
        self.assertTrue("get_sheet_" in globals())
```

```
if __name__ == "__main__":  
    main()
```

- Each test function must start with `test_`
- To check if function raises proper error:

```
with self.assertRaises(ValueError):  
    calc.divide(10, 0)
```

Setup and TearDown

- `setUp` is executed before every `test_function`.
- `tearDown` is executed after every `test_function`.

```
class TestEmployee(unittest.TestCase):  
  
    def setUp(self):  
        self.emp_1 = Employee('Corey', 'Schafer', 50000)  
        self.emp_2 = Employee('Sue', 'Smith', 60000)  
  
    def tearDown(self):  
        pass
```

- For classes run before and after the whole `TestCase`.
- Useful for cases when retrieving from database.

```
class TestEmployee(unittest.TestCase):  
  
    @classmethod  
    def setUpClass(cls):  
        print('setUpClass')  
  
    @classmethod  
    def tearDownClass(cls):  
        print('tearDownClass')
```

-

Testing in-between the functions

- Mocking.