**Basics of Unittesting**

### 1. ****Create a Test Class:****

* Tests are organized into classes that inherit from unittest.TestCase.
* Each test is a method within the class.

import unittest

class TestCalculator(unittest.TestCase):

def test\_add(self):

# Your test code goes here

pass

def test\_subtract(self):

# Your test code goes here

pass

### 2. ****Write Test Methods:****

* Create methods within the test class, starting with the word test.
* These methods will contain the actual test code.

def test\_add(self):

result = add(2, 3)

self.assertEqual(result, 5)

def test\_subtract(self):

result = subtract(5, 3)

self.assertEqual(result, 2)

### 3. ****Assertions:****

* Use various assertion methods provided by unittest to check conditions.
* Common assertions include assertEqual, assertTrue, assertFalse, assertRaises, etc.

def test\_add(self):

result = add(2, 3)

self.assertEqual(result, 5)

def test\_subtract(self):

result = subtract(5, 3)

self.assertEqual(result, 2)

### 4. ****Run Tests:****

* Use the unittest test runner to execute your tests.
* This can be done through the command line or an IDE.

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

### Example:

Let's say you have a simple calculator.py with the following functions:

# calculator.py

def add(a, b):

return a + b

def subtract(a, b):

return a - b

Now, you can create a test file test\_calculator.py:

# test\_calculator.py

import unittest

from calculator import add, subtract

class TestCalculator(unittest.TestCase):

def test\_add(self):

result = add(2, 3)

self.assertEqual(result, 5)

def test\_subtract(self):

result = subtract(5, 3)

self.assertEqual(result, 2)

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

### Running the Tests:

Save both calculator.py and test\_calculator.py.

Open a terminal in the directory containing the files.

***Run the following command:***

python test\_calculator.py

This will execute the tests, and you should see an output indicating whether the tests passed or failed.

Remember to replace the add and subtract functions in the example with your actual code. The idea is to test each function or method in isolation to ensure they behave as expected.