

ASSIGNMENT-2-TEST FIRST

SUBJECT: **SW555 (Agile Methods for Software Development)**

TEAM MEMBERS(GROUP 5):-

SAHIL DINESHBHAI PAMBHAR

SARTHAK JAIN

KENNETH JONES

PROFESSOR:- HARUN GULTEKIN

USER STORY

User Story Brief:

As a user, I want to utilize a comprehensive calculator application that supports a diverse set of operations, ranging from basic arithmetic to advanced functionalities. The calculator should include:

1. Basic Arithmetic Operations:

- Addition
- Multiplication
- Division

2. Logarithmic Functionality:

- Logarithm (base 10)

3. Trigonometric Functionality:

- Tangent with rounding to the nearest integer

4. Statistical Calculation:

- Median

5. Exponentiation:

- Calculation of the exponentiation of a positive base number by a positive exponent

6. Square Root Operation:

- Calculation of the square root of a positive number

These features aim to provide a comprehensive and versatile calculator experience, catering to both basic and advanced mathematical needs.

MANUAL TEST CODES

Feature: Comprehensive Calculator Operations

1. Scenario: Perform basic arithmetic operations
Given I open the comprehensive calculator application
When I input two positive numbers, for example, 5 and 7
And I click on the addition button
Then the result displayed should be 12

2. Scenario: Perform additional basic arithmetic operations
Given I open the comprehensive calculator application
When I input two numbers, for example, 10 and 3
And I click on the multiplication button
Then the result displayed should be 30

3. Scenario: Perform division with non-zero denominator
Given I open the comprehensive calculator application
When I input two positive numbers, for example, 15 and 3
And I click on the division button
Then the result displayed should be 5

4. Scenario: Perform logarithmic function
Given I open the comprehensive calculator application
When I input a positive number, for example, 100
And I click on the logarithm (base 10) button
Then the result displayed should be 2

5. Scenario: Perform advanced trigonometric function
Given I open the comprehensive calculator application
When I input an angle in degrees, for example, 45
And I click on the tangent button
Then the result displayed should be 1

6. Scenario: Perform statistical calculation
Given I open the comprehensive calculator application
When I input a list of numbers, for example, 2, 4, 6, 8, 10
And I click on the median button
Then the result displayed should be 6

7. Scenario: Perform exponentiation
Given I open the comprehensive calculator application
When I input a positive base number, for example, 2
And I input a positive exponent, for example, 4
And I click on the exponentiation button
Then the result displayed should be 16

8. Scenario: Perform square root

Given I open the comprehensive calculator application

When I input a positive number, for example, 36

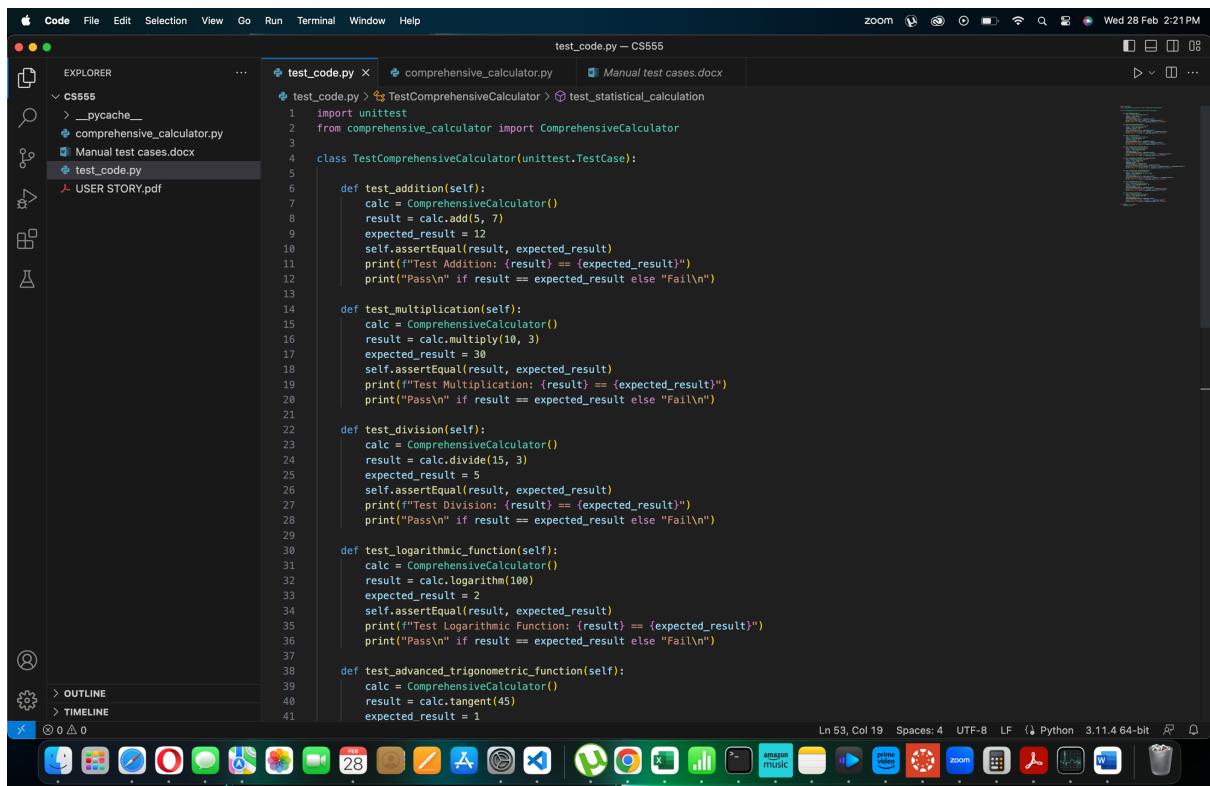
And I click on the square root button

Then the result displayed should be 6

THE CODE FOR THE USER STORY

```
1 import math
2 from statistics import mean, median
3
4 class ComprehensiveCalculator:
5     def add(self, x, y):
6         return x + y
7
8     def multiply(self, x, y):
9         return x * y
10
11    def divide(self, x, y):
12        if y != 0:
13            return x / y
14        else:
15            raise ValueError("Cannot divide by zero.")
16
17    def logarithm(self, x):
18        return math.log10(x)
19
20    def tangent(self, angle):
21        return math.tan(math.radians(angle))
22
23    def median(self, numbers):
24        return median(numbers)
25
26    def exponentiation(self, x, y):
27        return x ** y
28
29    def square_root(self, x):
30        return math.sqrt(x)
```

THE TEST CODE



```
test_code.py - CS555
import unittest
from comprehensive_calculator import ComprehensiveCalculator

class TestComprehensiveCalculator(unittest.TestCase):
    def test_addition(self):
        calc = ComprehensiveCalculator()
        result = calc.add(5, 7)
        expected_result = 12
        self.assertEqual(result, expected_result)
        print("Test Addition: {result} == {expected_result}")
        print("Pass\n" if result == expected_result else "Fail\n")

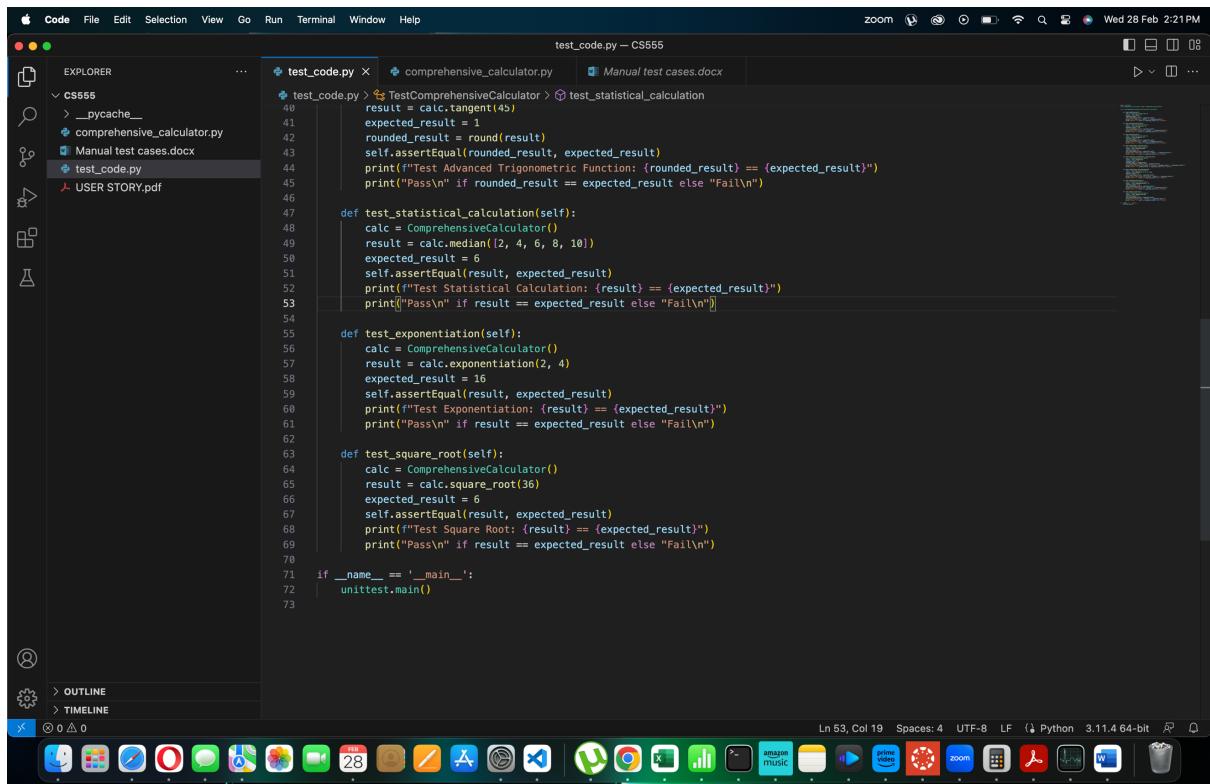
    def test_multiplication(self):
        calc = ComprehensiveCalculator()
        result = calc.multiply(10, 3)
        expected_result = 30
        self.assertEqual(result, expected_result)
        print("Test Multiplication: {result} == {expected_result}")
        print("Pass\n" if result == expected_result else "Fail\n")

    def test_division(self):
        calc = ComprehensiveCalculator()
        result = calc.divide(15, 3)
        expected_result = 5
        self.assertEqual(result, expected_result)
        print("Test Division: {result} == {expected_result}")
        print("Pass\n" if result == expected_result else "Fail\n")

    def test_logarithmic_function(self):
        calc = ComprehensiveCalculator()
        result = calc.logarithm(100)
        expected_result = 2
        self.assertEqual(result, expected_result)
        print("Test Logarithmic Function: {result} == {expected_result}")
        print("Pass\n" if result == expected_result else "Fail\n")

    def test_advanced_trigonometric_function(self):
        calc = ComprehensiveCalculator()
        result = calc.tangent(45)
        expected_result = 1
        self.assertEqual(result, expected_result)
        print("Test Advanced Trigonometric Function: {result} == {expected_result}")
        print("Pass\n" if result == expected_result else "Fail\n")

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



```
test_code.py - CS555
import unittest
from comprehensive_calculator import ComprehensiveCalculator

class TestComprehensiveCalculator(unittest.TestCase):
    def test_tangent(self):
        calc = ComprehensiveCalculator()
        result = calc.tangent(45)
        expected_result = 1
        rounded_result = round(result)
        self.assertEqual(rounded_result, expected_result)
        print("Test Tangent: {rounded_result} == {expected_result}")
        print("Pass\n" if rounded_result == expected_result else "Fail\n")

    def test_statistical_calculation(self):
        calc = ComprehensiveCalculator()
        result = calc.median([2, 4, 6, 8, 10])
        expected_result = 6
        self.assertEqual(result, expected_result)
        print("Test Statistical Calculation: {result} == {expected_result}")
        print("Pass\n" if result == expected_result else "Fail\n")

    def test_exponentiation(self):
        calc = ComprehensiveCalculator()
        result = calc.exponentiation(2, 4)
        expected_result = 16
        self.assertEqual(result, expected_result)
        print("Test Exponentiation: {result} == {expected_result}")
        print("Pass\n" if result == expected_result else "Fail\n")

    def test_square_root(self):
        calc = ComprehensiveCalculator()
        result = calc.square_root(36)
        expected_result = 6
        self.assertEqual(result, expected_result)
        print("Test Square Root: {result} == {expected_result}")
        print("Pass\n" if result == expected_result else "Fail\n")

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

TEST RESULTS

A screenshot of a macOS desktop environment showing a terminal window within the VS Code application. The terminal window is titled "test_code.py - CS555". It displays the output of a Python unittest run. The output shows various test cases being executed and passed, including tests for addition, trigonometric functions, division, exponentiation, logarithmic functions, multiplication, square root, and statistical calculations. The terminal also shows the command used to run the test code and the user's login information.

```
test_code.py x comprehensive_calculator.py Manual test cases.docx
test_code.py > TestComprehensiveCalculator > test_statistical_calculation
  1   import unittest
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
/usr/local/bin/python3 /Users/sahilpambhar/university/CS555/test_code.py
(base) sahilpambhar@sahils CS555 % /usr/local/bin/python3 /Users/sahilpambhar/university/CS555/test_code.py
Test Addition: 12 == 12
Pass
.Test Advanced Trigonometric Function: 1 == 1
Pass
.Test Division: 5.0 == 5
Pass
.Test Exponentiation: 16 == 16
Pass
.Test Logarithmic Function: 2.0 == 2
Pass
.Test Multiplication: 30 == 30
Pass
.Test Square Root: 6.0 == 6
Pass
.Test Statistical Calculation: 6 == 6
Pass
.
Ran 8 tests in 0.000s
OK
(base) sahilpambhar@sahils CS555 %
```