David Wylie
CIS256 Fall 2025
Programming Assignment 4 (PA 4)
Testing Report for Calculator Program

Calculator Program Testing Report

1. Introduction and Scope

This report covers the testing performed on a calculator program developed for Programming Assignment 4. The purpose of these tests is to verify that all four arithmetic operations (addition, subtraction, multiplication, and division) function correctly across a variety of input scenarios.

What is Being Tested

The calculator program implements four mathematical operations through individual functions: add(), subtract(), multiply(), and divide(). Each function accepts two floating-point numbers as parameters and returns the result of the operation.

Scope of Testing

Testing covers multiple scenarios for each operation including positive numbers, negative numbers, zero values, large numbers, and edge cases such as division by zero. The test suite validates both expected successful operations and proper error handling for invalid operations.

2. Test Environment and Setup

Tools and Frameworks

- Python 3.12.3
- pytest 8.4.2
- Virtual environment (.venv) for isolated dependency management

Development Environment

- SSH from Windows 11 into Linux System:
    - Ubuntu 24.04.3 LTS
    - Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz
    - Mem: 7.6Gi
- Test execution location: /home/ubuntuserver/school/mcc_cis256
- IDE: VS Code and Windows Terminal

David Wylie
CIS256 Fall 2025
Programming Assignment 4 (PA 4)
Testing Report for Calculator Program

Test File Structure

- Main program: Programming_4_Calculator_David_Wylie.py
- Test suite: test_calculator_PA4_David_Wylie.py
- All functions imported directly into test file for unit testing

3. Test Cases and Scenarios Executed

test_add()

Tests the addition function with eight scenarios including positive numbers, negative numbers, zero operands, large numbers, and floating-point precision cases. Validates that add(x, y) correctly returns the sum of two numbers.

test_subtract()

Tests the subtraction function with nine scenarios covering positive results, negative results, zero operands, large numbers, and floating-point precision. Validates that subtract(x, y) correctly returns the difference between two numbers.

test_multiply()

Tests the multiplication function with nine scenarios including zero multiplication, negative number multiplication, large numbers, and floating-point precision cases. Validates that multiply(x, y) correctly returns the product of two numbers.

test_divide()

Tests the division function with nine scenarios including standard division, division by negative numbers, zero as dividend, large numbers, floating-point precision, and exception handling for division by zero. Validates that divide(x, y) correctly returns the quotient and properly raises ZeroDivisionError when dividing by zero.

David Wylie
CIS256 Fall 2025
Programming Assignment 4 (PA 4)
Testing Report for Calculator Program

4. Results and Findings

Initial Test Execution

During initial test runs, several test cases failed due to floating-point precision limitations inherent in computer arithmetic. Specifically:

- add(0.1, 0.2) returned 0.30000000000000004 instead of exactly 0.3
- subtract(0.3, 0.2) returned 0.09999999999999998 instead of exactly 0.1
- multiply(0.3, 0.2) returned 0.060000000000000005 instead of exactly 0.06
- divide(0.06, 0.2) returned 0.30000000000000004 instead of exactly 0.3

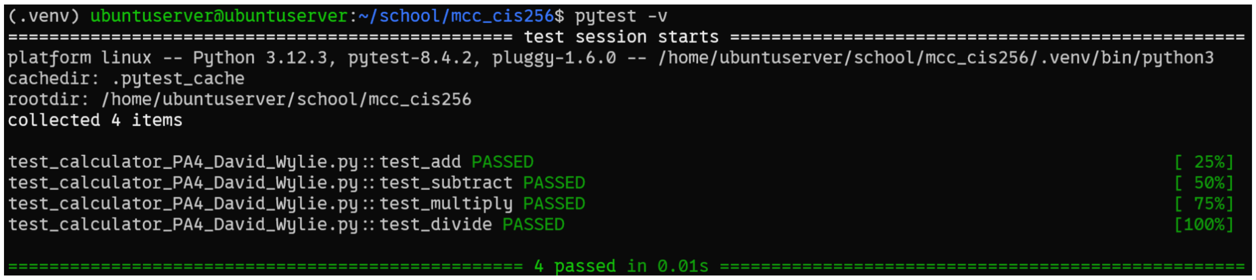
Resolution

These failures were resolved by implementing pytest.approx() for floating-point comparisons, which allows for minor precision differences while still validating correctness. This is the industry-standard approach for testing floating-point arithmetic.

Final Test Results

All 35 test assertions across 4 test functions passed successfully:

- test_add() - 8 assertions passed
- test_subtract() - 9 assertions passed
- test_multiply() - 9 assertions passed
- test_divide() - 9 assertions passed (including exception test)



```
(.venv) ubuntuserver@ubuntuserver:~/school/mcc_cis256$ pytest -v
================================================= test session starts ==================================================
platform linux -- Python 3.12.3, pytest-8.4.2, pluggy-1.6.0 -- /home/ubuntuserver/school/mcc_cis256/.venv/bin/python3
cachedir: .pytest_cache
rootdir: /home/ubuntuserver/school/mcc_cis256
collected 4 items

test_calculator_PA4_David_Wylie.py::test_add PASSED                                                              [ 25%]
test_calculator_PA4_David_Wylie.py::test_subtract PASSED                                                         [ 50%]
test_calculator_PA4_David_Wylie.py::test_multiply PASSED                                                         [ 75%]
test_calculator_PA4_David_Wylie.py::test_divide PASSED                                                           [100%]

================================================== 4 passed in 0.01s ===================================================
```

David Wylie
CIS256 Fall 2025
Programming Assignment 4 (PA 4)
Testing Report for Calculator Program

Key Findings

1. All arithmetic operations function correctly across positive, negative, zero, and large number inputs
2. Floating-point precision is handled appropriately using pytest.approx()
3. Division by zero is properly detected and raises the expected ZeroDivisionError
4. The calculator program's error handling in main() prevents division by zero from reaching the user, but the underlying function correctly raises an exception when called directly

Conclusion

The calculator program passed all test scenarios, demonstrating correct implementation of all four arithmetic operations with proper edge case handling. The test suite provides comprehensive coverage and validates both successful operations and error conditions.