Code Coverage Assignment Documentation

# 1. Program Development

Program Description:  
I created a simple Python calculator program that performs basic arithmetic operations. The program defines four distinct functions:  
- add(a, b): Returns the sum of two numbers.  
- subtract(a, b): Returns the difference.  
- multiply(a, b): Returns the product.  
- divide(a, b): Returns the quotient. It raises an exception if the divisor is zero.

## Source Code (calculator.py):

def add(a, b):  
 """Return the sum of a and b."""  
 return a + b  
  
def subtract(a, b):  
 """Return the difference of a and b."""  
 return a - b  
  
def multiply(a, b):  
 """Return the product of a and b."""  
 return a \* b  
  
def divide(a, b):  
 """Return the quotient of a divided by b. Raises a ValueError if b is zero."""  
 if b == 0:  
 raise ValueError("Cannot divide by zero.")  
 return a / b

# 2. Write Partial Unit Tests

a) Functions Tested and Why:

I chose to test the `add` and `divide` functions. `add` is the simplest to validate basic correctness, while `divide` includes exception handling which is crucial to verify.

## Test Code (test\_calculator.py):

import unittest  
from calculator import add, divide  
  
class TestCalculator(unittest.TestCase):  
 def test\_add(self):  
 self.assertEqual(add(2, 3), 5)  
 self.assertEqual(add(-1, 1), 0)  
  
 def test\_divide(self):  
 self.assertEqual(divide(10, 2), 5)  
 with self.assertRaises(ValueError):  
 divide(5, 0)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

# 3. Measure Code Coverage

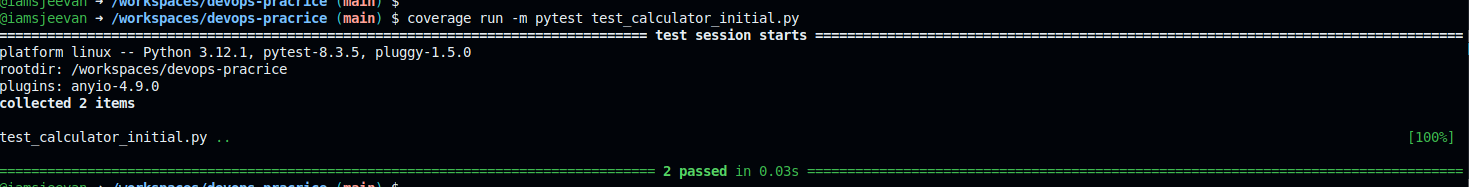
a) Tool Used: coverage.py

b) Installation and Usage:

- Installed using: pip install coverage  
- Run tests with coverage: coverage run -m unittest test\_calculator.py  
- Generate report: coverage report or coverage html

c) Initial Code Coverage: 50% (2 out of 4 functions tested)

d) Screenshot of Initial Report:



# 4. Improve Coverage

a) Additional Tests Added: `subtract` and `multiply` functions.

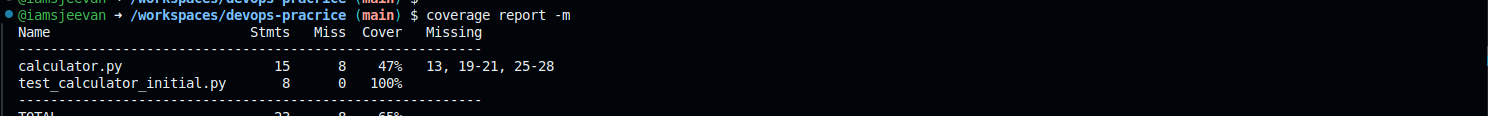
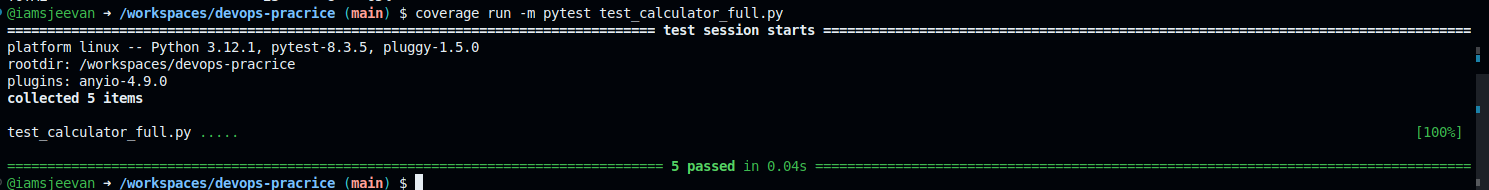
## Updated Test Code:

class TestCalculator(unittest.TestCase):  
 def test\_add(self):  
 self.assertEqual(add(2, 3), 5)  
 self.assertEqual(add(-1, 1), 0)  
  
 def test\_subtract(self):  
 self.assertEqual(subtract(5, 3), 2)  
 self.assertEqual(subtract(0, 4), -4)  
  
 def test\_multiply(self):  
 self.assertEqual(multiply(4, 5), 20)  
 self.assertEqual(multiply(0, 100), 0)  
  
 def test\_divide(self):  
 self.assertEqual(divide(10, 2), 5)  
 with self.assertRaises(ValueError):  
 divide(5, 0)

b) Re-run Coverage Tool:  
coverage run -m unittest test\_calculator.py  
coverage report

c) Final Code Coverage: 100%

d) Screenshot of Final Report:



# Submission Checklist

✔ Source code of your program  
  
✔ Initial test cases  
  
✔ Code coverage report before and after improvements  
  
✔ Final set of test cases  
  
✔ A brief explanation for each step