School of Computer Science and Artificial Intelligence

Lab Assignment # 8.2

Program : B. Tech (CSE)

Specialization : AIML

Course Title : AI Assisted Coding Course Code:

23CS002PC304

Semester : VI

Academic Session : 2025-2026

Name of Student : A. Prashanth

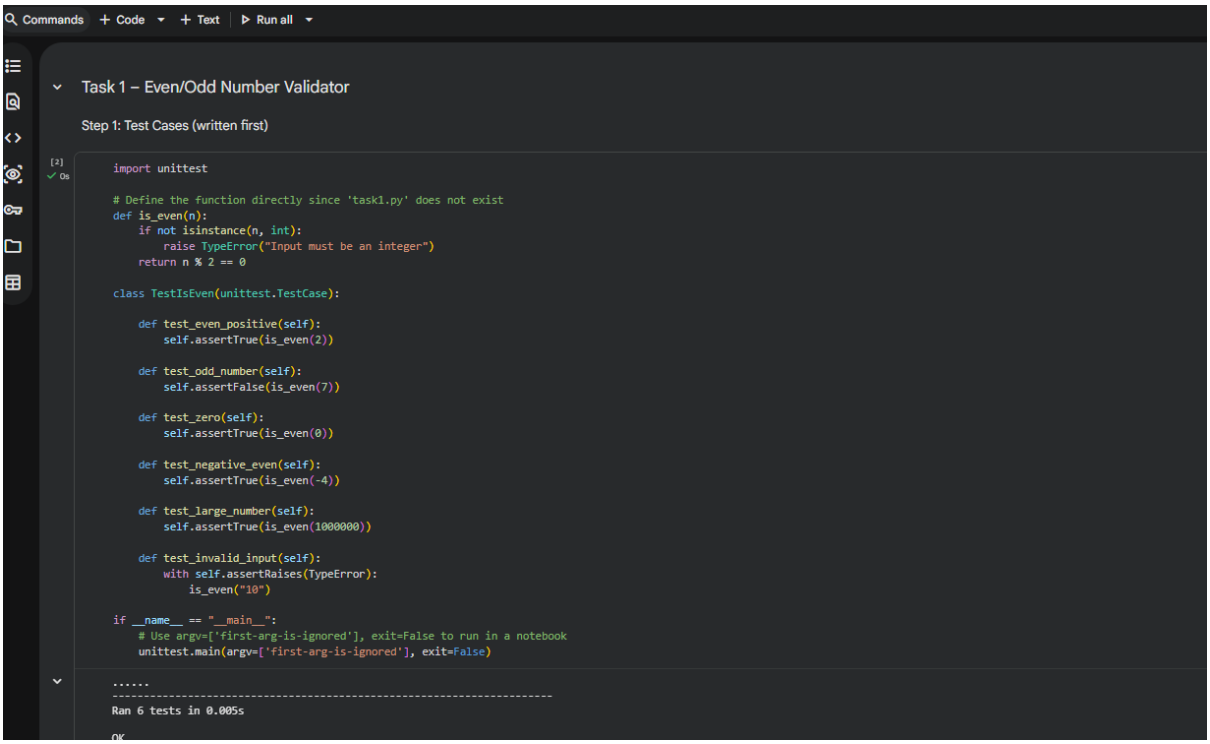
Enrollment No. : 2303A52092

Batch No. : 33

Date : 10/02/26

Task 1 – Even/Odd Number Validator

Step 1: Test Cases (written first)



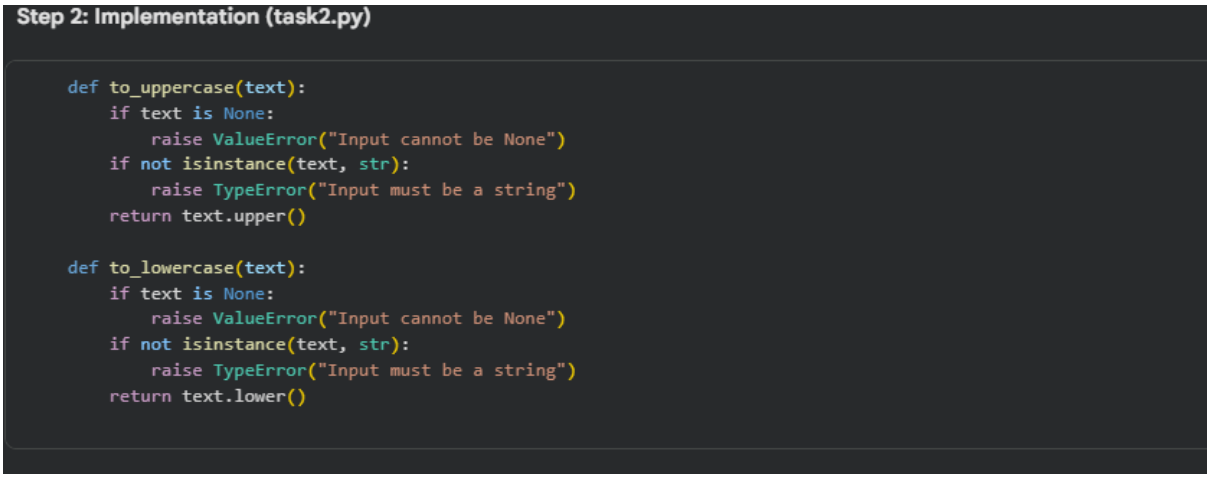
```
Commands + Code + Text ▷ Run all

Task 1 – Even/Odd Number Validator

Step 1: Test Cases (written first)

[2]
✓ 0s
    import unittest

    # Define the function directly since 'task1.py' does not exist
    def is_even(n):
        if not isinstance(n, int):
            raise TypeError("Input must be an integer")
        return n % 2 == 0

    class TestIsEven(unittest.TestCase):

        def test_even_positive(self):
            self.assertTrue(is_even(2))

        def test_odd_number(self):
            self.assertFalse(is_even(7))

        def test_zero(self):
            self.assertTrue(is_even(0))

        def test_negative_even(self):
            self.assertTrue(is_even(-4))

        def test_large_number(self):
            self.assertTrue(is_even(1000000))

        def test_invalid_input(self):
            with self.assertRaises(TypeError):
                is_even("10")

    if __name__ == "__main__":
        # Use argv=['first-arg-is-ignored'], exit=False to run in a notebook
        unittest.main(argv=['first-arg-is-ignored'], exit=False)

    ......
    ----------------------------------------------------------------------
    Ran 6 tests in 0.005s

    OK
```

Step 2: Implementation (task1.py)

```
def is_even(n):
    if not isinstance(n, int):
        raise TypeError("Input must be an integer")
    return n % 2 == 0
```

Task 2 – String Case Converter

Step 1: Test Cases

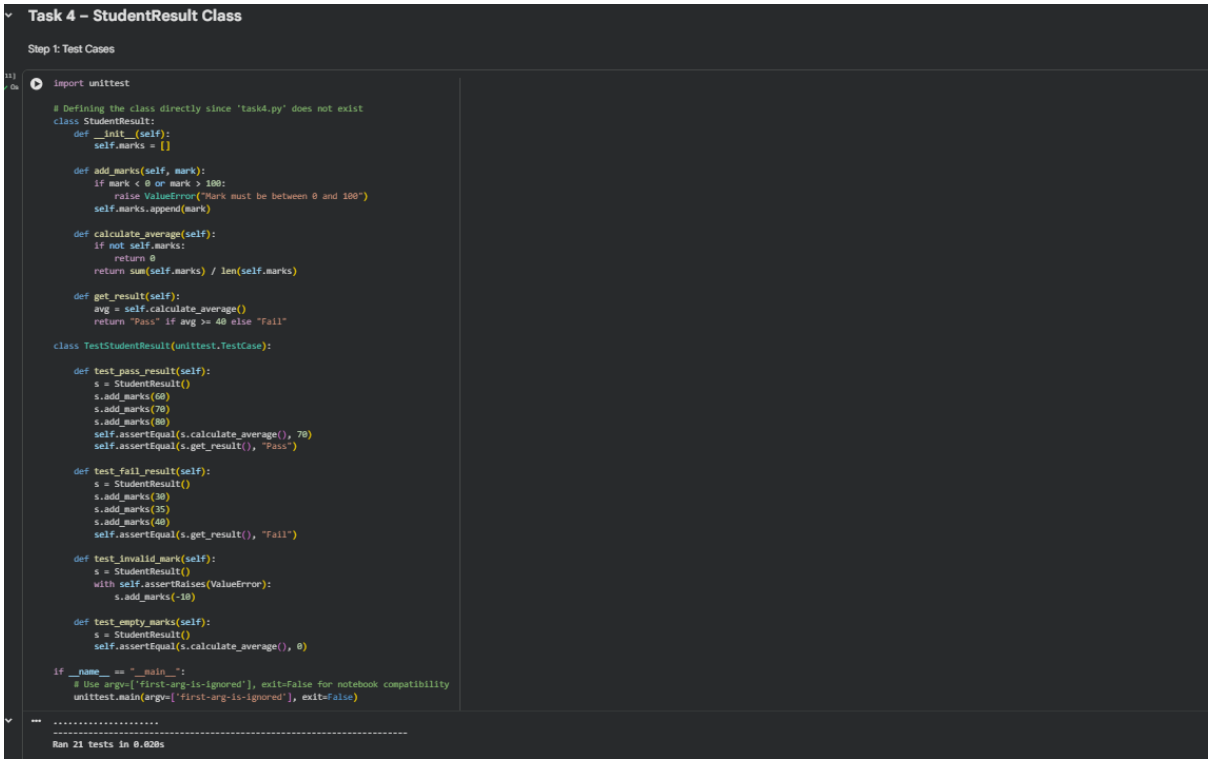


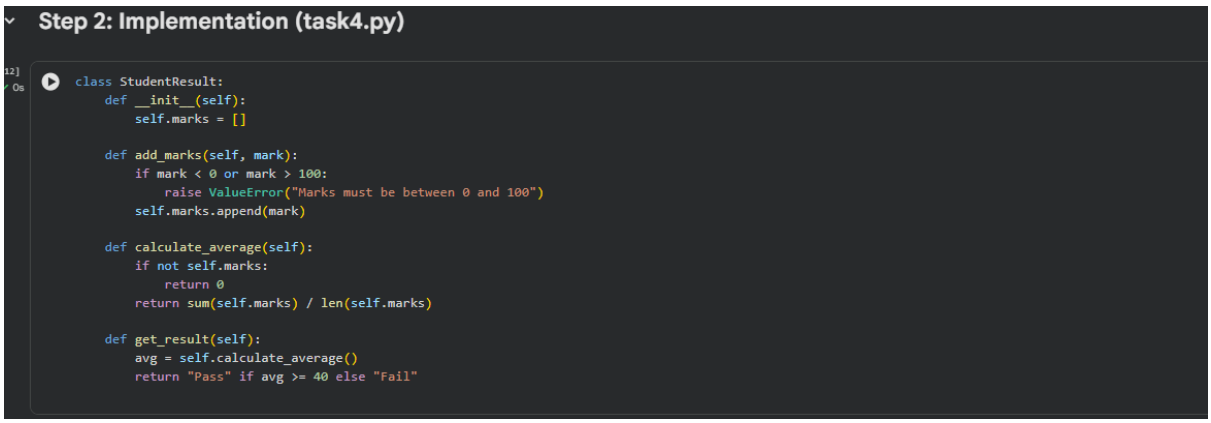
Task 3 – List Sum Calculator

Step 1: Test Cases

Task 4 – Student Result Class

Step 1: Test Cases





Task 5 – Username Validator

Step 1: Test Cases

Task 5 – Username Validator

Step 1: Test Cases

```
[14]
✓ 0s
import unittest

# Defining the function directly since 'task5.py' does not exist
def is_valid_username(username):
    if not isinstance(username, str):
        return False
    if len(username) < 3:
        return False
    if not username.isalnum():
        return False
    return True

class TestUsername(unittest.TestCase):

    def test_valid_username(self):
        self.assertTrue(is_valid_username("user01"))

    def test_short_username(self):
        self.assertFalse(is_valid_username("ai"))

    def test_space_in_username(self):
        self.assertFalse(is_valid_username("user name"))

    def test_special_characters(self):
        self.assertFalse(is_valid_username("user@123"))

    def test_non_string(self):
        self.assertFalse(is_valid_username(12345))

if __name__ == "__main__":
    # Use argv=['first-arg-is-ignored'], exit=False for notebook compatibility
    unittest.main(argv=['first-arg-is-ignored'], exit=False)
```

```
.......................
----------------------------------------------------------------------
Ran 26 tests in 0.027s

OK
```

Step 2: Implementation (task5.py)

```
[15]
✓ 0s
def is_valid_username(username):
    if not isinstance(username, str):
        return False
    if len(username) < 5:
        return False
    if " " in username:
        return False
    if not username.isalnum():
        return False
    return True
```

Lab Outcomes Covered

- Test cases written first (TDD style)
- Input validation & error handling
- Edge cases: empty, None, negative, large values
- unittest usage
- Clean and reliable implementations