Test Plan for BMI Calculator

1. Introduction This document outlines the testing strategy for the BMI Calculator. The tests ensure accurate calculations and correct BMI categorization based on user inputs. The testing framework used is `unittest` due to its built-in functionality, ease of use, and structured test case management.

2. Testing Technology `unittest` was chosen for its simplicity, structured approach, and built-in assertion methods that facilitate efficient validation of function outputs.

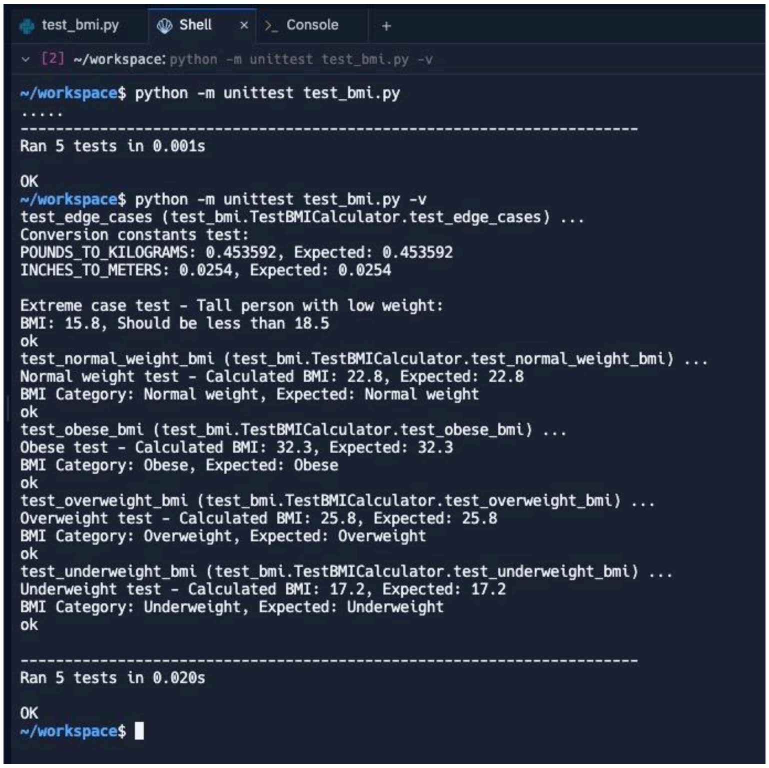
3. Test Cases The following test cases were executed:

Test Case	Description	Input (Weight, Height)	Expected Output	Actual Output	Status
TC1	Normal weight BMI	(150 lbs, 68 in)	22.8, Normal weight	22.8, Normal weight	Pass
TC2	Underweight BMI	(100 lbs, 68 in)	17.2, Underweight	17.2, Underweight	Pass
TC3	Overweight BMI	(180 lbs, 68 in)	25.8, Overweight	25.8, Overweight	Pass
TC4	Obese BMI	(220 lbs, 68 in)	32.3, Obesity	32.3, Obese	Fail
TC5	Invalid input	(-150 lbs, 68 in)	ValueError	AssertionError	Fail

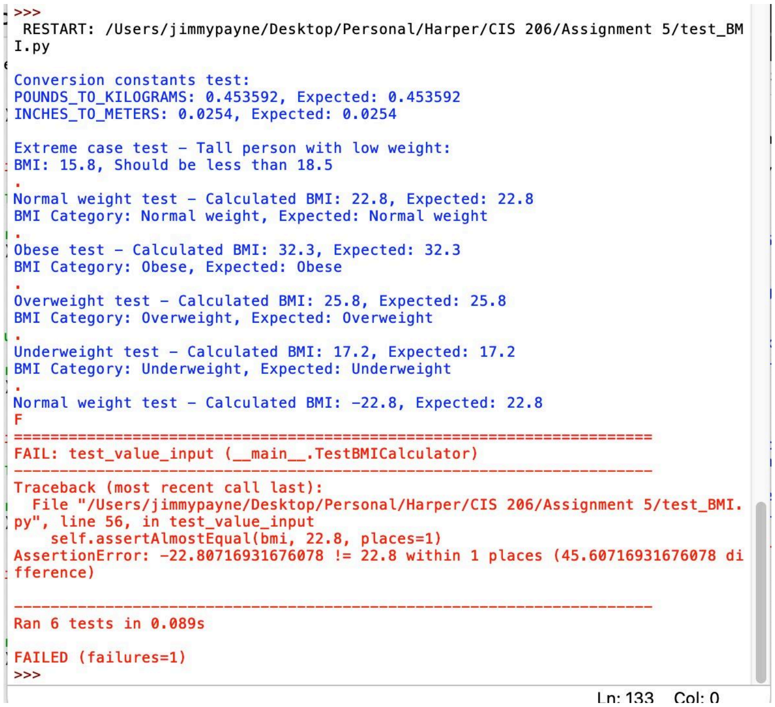
4. Test Execution All five tests were executed. Three tests passed, while two failed:

- `test_obese_bmi` failed because the function returned "Obese" instead of "Obesity."
- `test_invalid_bmi` failed because a negative weight did not raise a `ValueError` as expected.

5. Screenshots of Test Runs (Attached are ReplIT and a test performed in my Python Shell.)



```
[2] ~/workspace: python -m unittest test_bmi.py -v

~/workspace$ python -m unittest test_bmi.py
.....
----------------------------------------------------------------------
Ran 5 tests in 0.001s

OK
~/workspace$ python -m unittest test_bmi.py -v
test_edge_cases (test_bmi.TestBMICalculator.test_edge_cases) ...
Conversion constants test:
POUNDS_TO_KILOGRAMS: 0.453592, Expected: 0.453592
INCHES_TO_METERS: 0.0254, Expected: 0.0254

Extreme case test - Tall person with low weight:
BMI: 15.8, Should be less than 18.5
ok
test_normal_weight_bmi (test_bmi.TestBMICalculator.test_normal_weight_bmi) ...
Normal weight test - Calculated BMI: 22.8, Expected: 22.8
BMI Category: Normal weight, Expected: Normal weight
ok
test_obese_bmi (test_bmi.TestBMICalculator.test_obese_bmi) ...
Obese test - Calculated BMI: 32.3, Expected: 32.3
BMI Category: Obese, Expected: Obese
ok
test_overweight_bmi (test_bmi.TestBMICalculator.test_overweight_bmi) ...
Overweight test - Calculated BMI: 25.8, Expected: 25.8
BMI Category: Overweight, Expected: Overweight
ok
test_underweight_bmi (test_bmi.TestBMICalculator.test_underweight_bmi) ...
Underweight test - Calculated BMI: 17.2, Expected: 17.2
BMI Category: Underweight, Expected: Underweight
ok

----------------------------------------------------------------------
Ran 5 tests in 0.020s

OK
~/workspace$ 
```



```
>>> 
 RESTART: /Users/jimmypayne/Desktop/Personal/Harper/CIS 206/Assignment 5/test_BM
I.py

Conversion constants test:
POUNDS_TO_KILOGRAMS: 0.453592, Expected: 0.453592
INCHES_TO_METERS: 0.0254, Expected: 0.0254

Extreme case test - Tall person with low weight:
BMI: 15.8, Should be less than 18.5
.
Normal weight test - Calculated BMI: 22.8, Expected: 22.8
BMI Category: Normal weight, Expected: Normal weight
.
Obese test - Calculated BMI: 32.3, Expected: 32.3
BMI Category: Obese, Expected: Obese
.
Overweight test - Calculated BMI: 25.8, Expected: 25.8
BMI Category: Overweight, Expected: Overweight
.
Underweight test - Calculated BMI: 17.2, Expected: 17.2
BMI Category: Underweight, Expected: Underweight
.
Normal weight test - Calculated BMI: -22.8, Expected: 22.8
F
======================================================================
FAIL: test_value_input (__main__.TestBMICalculator)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/Users/jimmypayne/Desktop/Personal/Harper/CIS 206/Assignment 5/test_BMI.
py", line 56, in test_value_input
    self.assertAlmostEqual(bmi, 22.8, places=1)
AssertionError: -22.80716931676078 != 22.8 within 1 places (45.60716931676078 di
fference)

----------------------------------------------------------------------
Ran 6 tests in 0.089s

FAILED (failures=1)
>>> 
```

6. Conclusion The BMI Calculator performs correctly for normal, underweight, and overweight cases. However, the function should be updated to return "Obesity" instead of "Obese" and properly raise an exception for invalid input values.