**BMI Calculator – Test Report**

**Name:** Citlali Hernandez Lopez

**NetID:** ch3270

**GitHub username:** citlalih1421

1.Function Descriptions and Test Cases

**Function: calculate\_bmi(feet: int, inches: int, weight: float) -> tuple**

The calculate\_bmi function calculates the Body Mass Index (BMI) for a person based on their height (in feet and inches) and weight (in pounds). The function first converts the height from feet and inches to meters and weight from pounds to kilograms. Using the formula:

BMI = (weight in kilograms) / (height in meters)2

The result is then rounded to one decimal place. The function categorizes the BMI into one of four categories: “Underweight,” “Normal weight,” “Overweight,” or “Obese.”

**Tests:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Input** | **Expected Output** | **Explanation** |
| test\_underweight | 5ft 6in, 110 lbs | (17.8, Underweight) | This test checks the “Underweight” category, where the BMI is less than 18.5. |
| test\_normal\_weight | 5ft 8 in, 150lbs | (22.8, Normal Weight) | This test checks the “Normal weight” category, where the BMI is between 18.5 and 24.9. |
| test\_overweight | 5ft 10 in, 180lbs | (25.8, Overweight) | This test checks the “Overweight” category, where the BMI is between 25.0 and 29.9. |
| test\_obese | 6 ft, 250lbs | (33.9, Obese) | This test checks the “Obese” category, where the BMI is 30.0 or greater. |
| test\_boundary\_underweight | 5ft 8 in, 122lbs | (18.5, Normal Weight) | Tests the boundary between “Underweight” and “Normal weight” categories. |
| test\_boundary\_normal | 5ft 6 in, 154lbs | (24.9, Normal Weight) | Tests the boundary between “Normal weight” and “Overweight” categories. The BMI is just under 25.0. |
| test\_boundary\_overweight | 5ft 6 in, 185lbs | (29.9, Overweight) | Tests the boundary between “Normal weight” and “Overweight” categories. The BMI is just under 25.0. |

**Screenshot of unit test results with the following BMI function**

A screenshot of a computer program

AI-generated content may be incorrect.

2. Boundary Testing Technique

The **boundary testing technique** focuses on testing values that lie at or near the limits of the input domain. In the case of BMI categories, the boundaries are defined by the cut-off values for each category:

• “Underweight”: BMI < 18.5

• “Normal weight”: 18.5 ≤ BMI < 25.0

• “Overweight”: 25.0 ≤ BMI < 30.0

• “Obese”: BMI ≥ 30.0

In boundary testing, we examine the following:

• **Exact Boundary Values**: We check BMI values exactly at 18.5, 25.0, and 30.0 to ensure the function categorizes them correctly.

• **Above and Below Boundaries**: We also test values slightly below or above these boundaries (e.g., 18.4, 24.9, 29.9) to confirm the category changes at the appropriate point.

**Why this technique was chosen**: Boundary testing is essential for ensuring that category transitions are handled properly, and the function is robust at critical points. These boundaries determine the user’s BMI classification, so testing at these points ensures correctness.

3. Inducing Boundary Shift

Induced boundary shift by 0.1 at the lower boundary of “Normal Weight”

**Screenshot of test results with the following boundary shift**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

4. Did Your Test Cases Catch This Boundary Shift?

• Test Case Analysis: The test cases should catch any boundary shift issues because they test values that are very close to the boundaries. For example, the test\_boundary\_underweight and test\_boundary\_normal test cases test BMI values around 18.5, which should trigger a proper categorization.

• Boundary Shift Catching: However, if rounding issues exist or if the calculation does not correctly classify a BMI near the boundary (e.g., 18.4 vs 18.5), the test case might fail to catch the issue. If a shift happens, the function may incorrectly categorize a person as “Underweight” when they should be “Normal weight” or vice versa.

Why Test Cases May Fail:

• If the BMI calculation or rounding is not handled correctly, these boundary tests might not yield the correct categories.

• For example, the difference between 18.4 and 18.5 might not be significant due to rounding, but it may lead to a category mismatch.

5. Setup and Execution Instructions

Setup Instructions:

• Operating System: Windows 10

• Python Version: Python 3.x (Ensure Python is installed on your machine)

• Required Libraries: unittest (comes with Python standard library)

Steps to Execute:

1. Clone or download the repository:

Link to download the from GitHub repository: <https://github.com/citlalih1421/cse4283_assignment2> .

2. Install Python 3.x on your Windows 10 system, if not already installed.

Link to download python:

<https://www.python.org/getit/>

3. Navigate to the directory where bmi\_calculator.py, test\_bmi\_calculator.py, and cli.py are stored.

`cd <directory>`

4. Open a command prompt or PowerShell window and run:

`python3 -m unittest unit\_test.py`

5. The tests will execute, and you will see the results in the console output.

6. Run the CLI app:

`python3 cli.py`

6. Screenshots

Underweight:

A screenshot of a computer program

AI-generated content may be incorrect.

Normal Weight:

A screenshot of a computer program

AI-generated content may be incorrect.

Overweight:

A screenshot of a computer

AI-generated content may be incorrect.

Obese:

A screenshot of a computer

AI-generated content may be incorrect.