**Discussion Forum Unit 4**

### **1. Precondition Violation:**

A pre-condition is a condition that must be true before a function is called. If the pre-condition is not satisfied, the function may not work as expected. In my example of checking fuel levels in a car, the precondition is that the fuel level should be a non-negative value.

**Example:**

***Code:***

def check\_fuel\_level(fuel\_level):

# Check if the fuel level is valid (non-negative).

if fuel\_level < 0:

raise ValueError(“Fuel level cannot be negative.")

# usecase

try:

check\_fuel\_level(-10)

except ValueError as e:

print(f"Error: {e}")

Output:

Error: Fuel level cannot be negative.

### **2. Postcondition Violation:**

A post-condition is a condition that should be true after the function is done with its work. If the post-condition is not met, it indicates that the function did not perform as expected. In my example of checking fuel levels, a postcondition might be that the function returns a boolean indicating whether the fuel level is above a certain threshold.

**Example:**

def is\_fuel\_level\_sufficient(fuel\_level, threshold=30):

# Check if the fuel level is above a specified threshold.

result = fuel\_level > threshold

if not result:

raise AssertionError("Postcondition violated: Fuel level is not above the threshold.")

return result

#Use-case

try:

result = is\_fuel\_level\_sufficient(25)

print(f"Is fuel level sufficient? {result}")

except AssertionError as e:

print(f"Error: {e}")

Output:

Error: Postcondition violated: Fuel level is not above the threshold.

### **3. Return Value or Usage Issue:**

This occurs when the function executes without any errors, but the return value or the way it is used doesn't meet expectations. For instance, if a function is expected to return a boolean but returns an integer, it might lead to issues.

**Example:**

**Code:**

def is\_fuel\_level\_high(fuel\_level, threshold=50):

# Check if the fuel level is high.

return fuel\_level > threshold

# Incorrect usage (expecting a boolean, but using it as an integer)

result = is\_fuel\_level\_high(60) + 5

print(f"Result: {result}")

**Output:**

Result: 6

***REFERENCES:***

Downey, A. B. (2015). Think Python: How to Think Like a Computer Scientist (version: 2.2.23). Green Tea Press.