AI ASSISTED CODING

ASSIGNMENT-7.5

Name: Harshitha Guda

H.T.No: 2303A51102

Task 1 (Mutable Default Argument – Function Bug)

Task: Analyze given code where a mutable default argument causes unexpected behavior. Use AI to fix it.

Bug: Mutable default argument

def add_item(item, items=[]):

items.append(item)

return items

print(add_item(1))

print(add_item(2))

Expected Output: Corrected function avoids shared list bug.

```
1  def add_item(item, items=None):
2      if items is None:
3          items = []
4      items.append(item)
5      print(items)
6  add_item(1)
7  add_item(2)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
[1]
[2]
PS C:\Users\gudah>
```

Task 2 (Floating-Point Precision Error)

Task: Analyze given code where floating-point comparison fails. Use AI to correct with tolerance.

Bug: Floating point precision issue

def check_sum():

return (0.1 + 0.2) == 0.3

print(check_sum())

Expected Output: Corrected function

```
10  import math
11  def check_sum():
12      return math.isclose(0.1 + 0.2, 0.3)
13  print(check_sum())
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
True
PS C:\Users\gudah>
```

Task 3 (Recursion Error – Missing Base Case)

Task: Analyze given code where recursion runs infinitely due to missing base case. Use AI to fix.

Bug: No base case

def countdown(n):

print(n)

return countdown(n-1)

countdown(5)

Expected Output : Correct recursion with stopping condition.

```
16  def countdown(n):
17      if n < 0:
18          return
19      print(n)
20      return countdown(n-1)
21  countdown(5)
22
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
5
4
3
2
1
0
```

Task 4 (Dictionary Key Error)

Task: Analyze given code where a missing dictionary key causes error. Use AI to fix it.

Bug: Accessing non-existing key

def get_value():

data = {"a": 1, "b": 2}

return data["c"]

print(get_value())

Expected Output: Corrected with .get() or error handling.

```
24  def get_value():
25      data = {"a": 1, "b": 2}
26      return data.get("c", "Key not found")
27  print(get_value())
28
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
Key not found
PS C:\Users\gudah>
```

Task 5 (Infinite Loop – Wrong Condition)

Task: Analyze given code where loop never ends. Use AI to detect and fix it.

Bug: Infinite loop

def loop_example():

i = 0

while i < 5:

print(i)

Expected Output: Corrected loop increments i.

```
30  def loop_example():
31      i = 0
32      while i < 5:
33          print(i)
34          i+=1
35  loop_example()
36
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
0
1
2
3
4
```

Task 6 (Unpacking Error – Wrong Variables)

Task: Analyze given code where tuple unpacking fails. Use AI to fix it.

Bug: Wrong unpacking

a, b = (1, 2, 3)

Expected Output: Correct unpacking or using _ for extra values.

```
37  a, b, *_ = (1, 2, 3)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
PS C:\Users\gudah>
```

Task 7 (Mixed Indentation – Tabs vs Spaces)

Task: Analyze given code where mixed indentation breaks execution. Use AI to fix it.

Bug: Mixed indentation

def func():

x = 5

y = 10

return x+y

Expected Output : Consistent indentation applied.

```
39  def func():
40      x = 5
41      y = 10
42      return x+y
43  print(func())
44
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
15
PS C:\Users\gudah>
```

Task 8 (Import Error – Wrong Module Usage)

Task: Analyze given code with incorrect import. Use AI to fix.

Bug: Wrong import

import maths

print(maths.sqrt(16))

Expected Output: Corrected to import math

```
45  import math
46  print(math.sqrt(16))
47
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
4.0
PS C:\Users\gudah>
```

Task 9 (Unreachable Code – Return Inside Loop)

Task: Analyze given code where a return inside a loop prevents full iteration. Use AI to fix it.

Bug: Early return inside loop

def total(numbers):

for n in numbers:

return n

print(total([1,2,3]))

Expected Output: Corrected code accumulates sum and returns after loop.

```
48  def total(numbers):
49      s=0
50      for n in numbers:
51          s+=n
52      return s
53  print(total([1,2,3]))
54
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
6
PS C:\Users\gudah>
```

Task 10 (Name Error – Undefined Variable)

Task: Analyze given code where a variable is used before being defined. Let AI detect and fix the error.

Bug: Using undefined variable

def calculate_area():

return length * width

print(calculate_area())

Requirements:

• Run the code to observe the error.

• Ask AI to identify the missing variable definition.

• Fix the bug by defining length and width as parameters.

• Add 3 assert test cases for correctness.

Expected Output :

• Corrected code with parameters.

• AI explanation of the bug.

Successful execution of assertions.

```
56  # Function to calculate the area of a rectangle
57  def calculate_area(length, width):
58      # Multiply length and width to get area
59      return length * width
60
61  # Call the function with length=5 and width=10, then print the result
62  print(calculate_area(5, 10))
63
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
50
PS C:\Users\gudah>
```

```python
from 2303A51102_Lab_Assignment_7_5 import calculate_area

def test_calculate_area():
    assert calculate_area(5, 10) == 50
    assert calculate_area(3, 4) == 12
    assert calculate_area(0, 5) == 0

test_calculate_area()
```

Task 11 (Type Error – Mixing Data Types Incorrectly)

Task: Analyze given code where integers and strings are added incorrectly. Let AI detect and fix the error.

Bug: Adding integer and string

def add_values():

return 5 + "10"

print(add_values())

Requirements:

• Run the code to observe the error.

• AI should explain why int + str is invalid.

• Fix the code by type conversion (e.g., int("10") or str(5)).

• Verify with 3 assert cases.

Expected Output #6:

• Corrected code with type handling.

• AI explanation of the fix.

Successful test validation.

```python
def add_values():
    # Indent the return statement to be inside the function body
    # Convert the string "10" to an integer before adding
    return 5 + int("10")

# Call the function and print the result
print(add_values())
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
15
PS C:\Users\gudah>
```

```python
from 2303A51102_Lab_Assignment_7_5 import add_values

def test_add_values():
    assert add_values() == 15
    assert add_values() == 5 + int("10")
    assert isinstance(add_values(), int)

test_add_values()
print("All tests passed!")
```

Task 12 (Type Error – String + List Concatenation)

Task: Analyze code where a string is incorrectly added to a list.

Bug: Adding string and list

def combine():

return "Numbers: " + [1, 2, 3]

print(combine())

Requirements:

• Run the code to observe the error.

• Explain why str + list is invalid.

• Fix using conversion (str([1,2,3]) or " ".join()).

• Verify with 3 assert cases.

Expected Output:

• Corrected code

• Explanation

• Successful test validation

```python
# str + list is invalid because Python cannot concatenate a string with a list directly
# Strings and lists are different types, and the + operator doesn't know how to combine them
# You must convert the list to a string first using str() or join()
def combine():
    # Fix: Convert list to string using str()
    return "Numbers: " + str([1, 2, 3])
print(combine())
# Verify with 3 assert cases
assert combine() == "Numbers: [1, 2, 3]", "Test 1 failed"
assert isinstance(combine(), str), "Test 2 failed"
assert "Numbers:" in combine(), "Test 3 failed"
print("All assertions passed!")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
15
PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
Numbers: [1, 2, 3]
All assertions passed!
PS C:\Users\gudah>
```

Task 13 (Type Error – Multiplying String by Float)

Task: Detect and fix code where a string is multiplied by a float.

Bug: Multiplying string by float

def repeat_text():

return "Hello" * 2.5

print(repeat_text())

Requirements:

• Observe the error.

• Explain why float multiplication is invalid for strings.

• Fix by converting float to int.

• Add 3 assert test cases

```
87  # str * float is invalid because Python cannot multiply a string by a float
88  # The * operator for strings only works with integers to repeat the string
89  # You must convert the float to an integer first using int()
90  def repeat_text():
91      # Fix: Convert float to int
92      return "Hello" * int(2.5)
93
94  print(repeat_text())
95
96  # Verify with 3 assert cases
97  assert repeat_text() == "HelloHello", "Test 1 failed"
98  assert isinstance(repeat_text(), str), "Test 2 failed"
99  assert len(repeat_text()) == 10, "Test 3 failed"
100 print("All assertions passed!")
101
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
HelloHello
All assertions passed!
PS C:\Users\gudah>
```

Task 14 (Type Error – Adding None to Integer)

Task: Analyze code where None is added to an integer.

Bug: Adding None and integer

def compute():

value = None

return value + 10

print(compute())

Requirements:

• Run and identify the error.

• Explain why NoneType cannot be added.

• Fix by assigning a default value.

• Validate using asserts.

```python
def compute():
    value = 0  # Assign a default value
    return value + 10

result = compute()
print(result)

# Validate using asserts
assert result == 10, "Test 1 failed"
assert isinstance(result, int), "Test 2 failed"
assert result > 0, "Test 3 failed"
print("All assertions passed!")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
10
All assertions passed!
PS C:\Users\gudah>
```

Task 15 (Type Error – Input Treated as String Instead of Number)

Task: Fix code where user input is not converted properly.

Bug: Input remains string

def sum_two_numbers():

a = input("Enter first number: ")

b = input("Enter second number: ")

return a + b

print(sum_two_numbers())

Requirements:

• Explain why input is always string.

• Fix using int() conversion.

• Verify with assert test cases.

```python
def sum_two_numbers():
    a = int(input("Enter first number: "))  # Convert input to int
    b = int(input("Enter second number: "))  # Convert input to int
    return a + b

result = sum_two_numbers()
print(result)

# Verify with assert test cases
assert isinstance(result, int), "Result should be an integer"
assert result == (int(input("Enter first number: ")) + int(input("Enter second number: "))), "Sum does not match expected value"
print("All assertions passed!")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\gudah> & C:/Python314/python.exe c:/Users/gudah/OneDrive/Documents/AIAC/2303A51102_Lab_Assignment_7.5.py
Enter first number: 10
Enter second number: 20
30
```