

ASSIGNMENT- 7.5

Name: POOJITHA.EDDE

HT.No: 2303A51356

Batch: 20

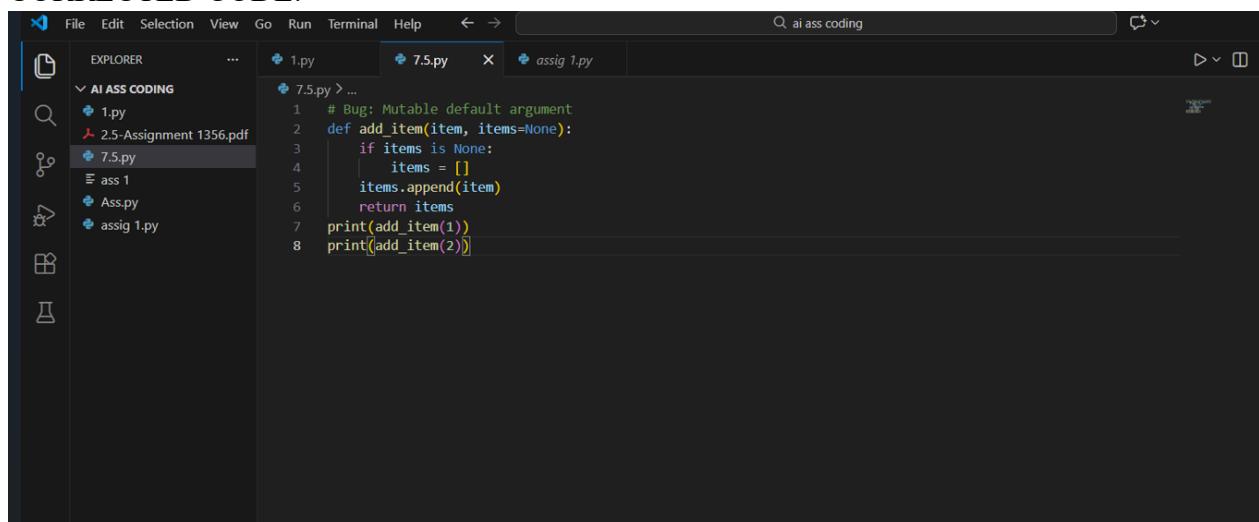
Task 1 (Mutable Default Argument – Function Bug)

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.

CORRECTED CODE:

A screenshot of the Visual Studio Code (VS Code) interface. The title bar says "ai ass coding". The left sidebar shows the "EXPLORER" view with a folder named "AI ASS CODING" containing files: 1.py, 2.5-Assignment 1356.pdf, 7.5.py, ass1, Ass.py, and assig 1.py. The main editor area has tabs for 1.py, 7.5.py (which is currently active), and assig 1.py. The code in 7.5.py is:

```
# Bug: Mutable default argument  
def add_item(item, items=None):  
    if items is None:  
        items = []  
    items.append(item)  
    return items  
print(add_item(1))  
print(add_item(2))
```

OUTPUT:

```
● PS C:\Users\pooji\OneDrive\Desktop\ai ass coding> & C:/Users/pooji/OneDrive/Desktop/ai ass coding/7.5.py"
[1]
[2]
○ PS C:\Users\pooji\OneDrive\Desktop\ai ass coding>
```

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

CORRECTED CODE:

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files in the 'AI ASS CODING' folder: 1.py, 7.5.py, ass 1, Ass.py, and assig 1.py. The Editor pane in the center displays the content of 7.5.py:

```
# Bug: Floating point precision issue
def check_sum():
    return abs((0.1 + 0.2) - 0.3) < 1e-10
print(check_sum())
```

OUTPUT:

The screenshot shows the VS Code terminal pane. The title bar indicates the Python environment. The terminal output shows the execution of 7.5.py:

```
PS C:\Users\pooji\OneDrive\Desktop\ai ass coding> & C:/Users/pooji/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/pooji/OneDrive/Desktop/ai ass coding/7.5.py"
True
PS C:\Users\pooji\OneDrive\Desktop\ai ass coding>
```

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.

CORRECTED CODE:

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files: 1.py, 2.5-Assignment 1356.pdf, 7.5.py, ass 1, Ass.py, and assig 1.py. The Editor pane on the right displays the following Python code:

```
# Bug: No base case
def countdown(n):
    if n <= 0:
        return
    print(n)
    return countdown(n-1)
countdown(5)
```

OUTPUT:

The screenshot shows the VS Code terminal window. The terminal tab is selected at the top. The output shows the execution of the script 7.5.py:

```
PS C:\Users\pooji\OneDrive\Desktop\ai ass coding> & c:/Users/pooji/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/pooji/OneDrive/Desktop/ai ass coding/7.5.py"
5
4
3
2
1
```

At the bottom of the terminal window, it says "Ln 7, Col 13 Spaces: 4 UTF-8 CRLF".

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.

CORRECTED CODE:

The screenshot shows a code editor interface with a dark theme. The left sidebar is titled 'EXPLORER' and lists several files: '1.py', '2.5-Assignment 1356.pdf', '7.5.py' (which is selected), 'ass 1', 'Ass.py', and 'assig 1.py'. The main area displays the contents of '7.5.py':

```
# Bug: Accessing non-existing key
def get_value():
    data = {"a": 1, "b": 2}
    return data.get("c", "Key not found")
print(get_value())
```

OUTPUT:

The screenshot shows the 'TERMINAL' tab of the code editor. It displays the command-line output of running the script:

```
PS C:\Users\pooji\OneDrive\Desktop\ai ass coding> & c:/users/pooji/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/pooji/OneDrive/Desktop/ai ass coding/7.5.py"
Key not found
PS C:\Users\pooji\OneDrive\Desktop\ai ass coding>
```

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.

CORRECTED CODE:

```
# Bug: Infinite loop
def loop_example():
    i = 0
    while i < 5:
        print(i)
        i += 1
loop_example()
```

OUTPUT:

```
PS C:\Users\pooji\OneDrive\Desktop\ai ass coding> & c:/Users/pooji/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/pooji/OneDrive/Desktop/ai ass coding/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.

CORRECTED CODE:

A screenshot of the Visual Studio Code interface. The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar at the top right. The Explorer sidebar shows files like 1.py, 2.5-Assignment 1356.pdf, 7.5.py, ass 1, Ass.py, and assig 1.py. The main editor tab is titled '7.5.py' and contains the following code:

```
# Bug: Wrong unpacking
a, b, c = (1, 2, 3)
```

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.

CORRECTED CODE:

A screenshot of the Visual Studio Code interface, similar to the previous one but with a more modern UI. The Explorer sidebar shows the same files. The main editor tab is titled '7.5.py' and contains the following corrected code:

```
# Bug: Mixed indentation
def func():
    x = 5
    y = 10
    return x+y
print(func())
```

OUTPUT:

A screenshot of the Visual Studio Code interface. The terminal tab is active, showing the following text:

```
PS C:\Users\pooji\OneDrive\Desktop\ai ass coding> & C:/Users/pooji/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/pooji/OneDrive/Desktop/ai ass coding/7.5.py"
15
○ PS C:\Users\pooji\OneDrive\Desktop\ai ass coding>
```

The status bar at the bottom indicates the file is 110.2 KB.

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

CORRECTED CODE:

A screenshot of the Visual Studio Code interface. The Explorer sidebar shows a folder named 'AI ASS CODING' containing files 1.py, 2.5-Assignment 1356.pdf, and 7.5.py. The 7.5.py file is open in the editor, displaying the following code:

```
#bug: Wrong import
import math
print(math.sqrt(16))
```

OUTPUT:

A screenshot of the Visual Studio Code interface. The terminal tab is active, showing the following text:

```
PS C:\Users\pooji\OneDrive\Desktop\ai ass coding> & C:/Users/pooji/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/pooji/OneDrive/Desktop/ai ass coding/7.5.py"
4.0
○ PS C:\Users\pooji\OneDrive\Desktop\ai ass coding>
```

The status bar at the bottom indicates the file is 107.4 KB.

