

AI ASSISTANT CODING

ASSIGNMENT-7

Name: Kommu Madhupriya

Hallticket:2303A51583

Batch:22

Lab 7: Error Debugging with AI: Systematic approaches to finding and fixing bugs

Lab Objectives:

- To identify and correct syntax, logic, and runtime errors in **Python programs using AI tools.**
- To understand common programming bugs and AI-assisted debugging suggestions.
- To evaluate how AI explains, detects, and fixes different types of coding errors.
- To build confidence in using AI to perform structured debugging practices.

Lab Outcomes (LOs):

After completing this lab, students will be able to:

- Use AI tools to detect and correct syntax, logic, and runtime errors.
- Interpret AI-suggested bug fixes and explanations.
- Apply systematic debugging strategies supported by AI-generated insights.

Refactor buggy code using responsible and reliable programming patterns.

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.

code:

The screenshot shows the Microsoft Visual Studio Code interface. In the top left, there are three tabs: 'priya.py', 'assi 1.py', and 'assi 7.py'. The 'assi 7.py' tab is active and contains the following Python code:

```
#Fix the Python function that uses a mutable default list causing shared values between function calls.  
def add_item(item, items=None):  
    if items is None:  
        items = [] # create a new list for each call  
    items.append(item)  
    return items  
print(add_item(1))  
print(add_item(2))
```

In the bottom right corner of the code editor, there is a status bar showing the file path 'C:\Users\kommu\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/kommu/Desktop/ai assisted coding/assi 7.py"', the line number 'Ln 6, Col 23', and the column number 'Spaces: 4'. The status bar also indicates the terminal is using 'UTF-8' encoding and 'CRLF' line endings, and shows the Python version '3.13.9 (Microsoft Store)'. The bottom right of the status bar shows the date and time '17-02-2026 13:05'.

Below the code editor, the terminal window displays the output of the script:

```
[1]  
[2]  
PS C:\Users\kommu\Desktop\ai assisted 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

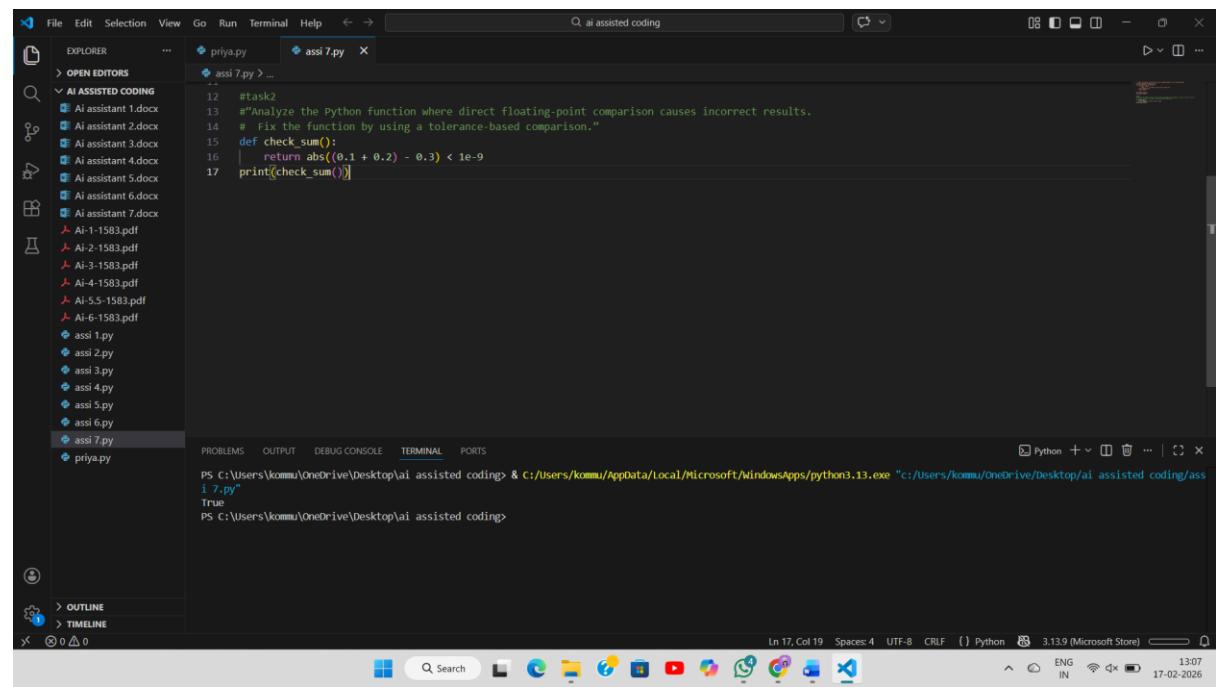
```
def check_sum():
```

```
    return (0.1 + 0.2) == 0.3
```

```
print(check_sum())
```

Expected Output: Corrected function

code:



The screenshot shows the Microsoft Visual Studio Code interface. The Explorer sidebar on the left lists various files, including several Microsoft Word documents and Python files like 'priya.py' and 'assi 7.py'. The main editor area contains the following Python code:

```
12 #task2
13 #Analyze the Python function where direct floating-point comparison causes incorrect results.
14 # Fix the function by using a tolerance-based comparison.
15 def check_sum():
16     return abs((0.1 + 0.2) - 0.3) < 1e-9
17 print(check_sum())
```

The terminal at the bottom shows the command line output:

```
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/users/kommu/appData/local/Microsoft/WindowsApps/python3.13.exe "c:/Users/kommu/OneDrive/Desktop/ai assisted coding/assi 7.py"
True
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>
```

The status bar at the bottom right indicates the file is 3.13.9 (Microsoft Store) and the date is 17-02-2026.

Task 3 (Recursion Error – Missing Base Case)

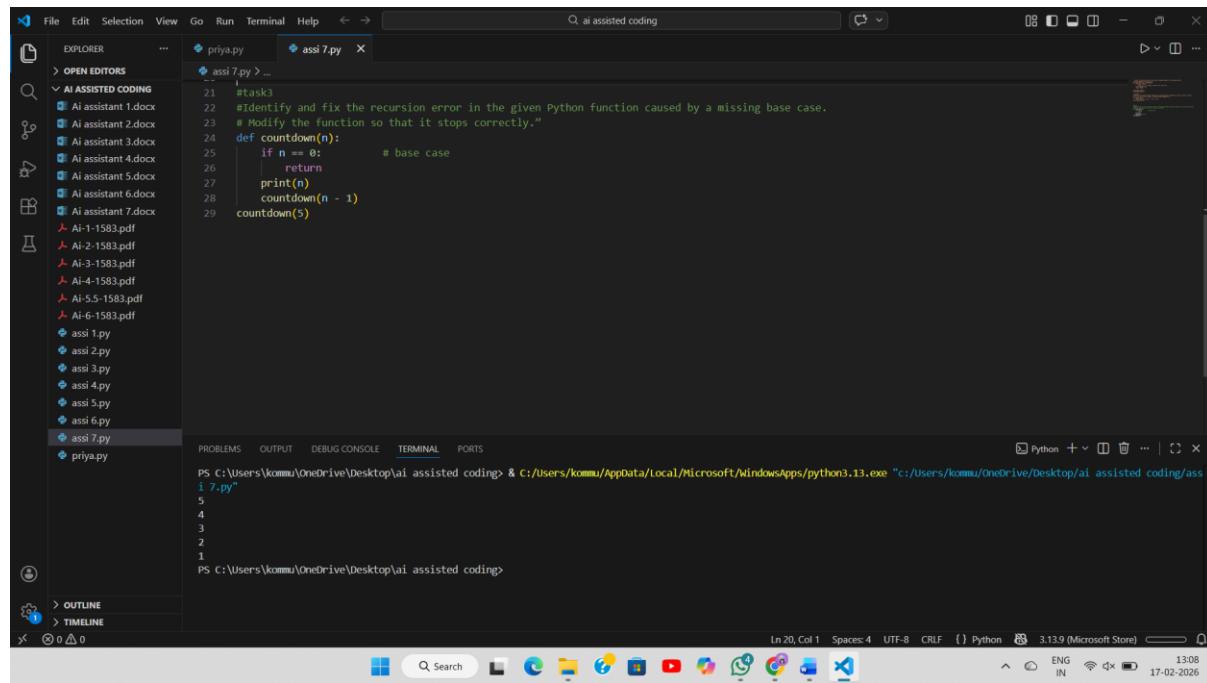
Task: Analyze given code where recursion runs infinitely due to missing base case. Use AI to fix.

Bug: No base case

```
def countdown(n):
    print(n)
    return countdown(n-1)

countdown(5)
```

Expected Output : Correct recursion with stopping condition.
code:



The screenshot shows the Microsoft Visual Studio Code interface. The code editor displays the following Python function:

```
21  #task3
22  #Identify and fix the recursion error in the given Python function caused by a missing base case.
23  # Modify the function so that it stops correctly."
24  def countdown(n):
25      if n == 0:          # base case
26          return
27      print(n)
28      countdown(n - 1)
29  countdown(5)
```

The terminal below the code editor shows the output of running the script:

```
PS C:\Users\kommu\Desktop\ai assisted coding> & c:/users/kommu/appData/local/Microsoft/WindowsApps/python3.13.exe "c:/Users/kommu/Desktop/ai assisted coding/ass1 7.py"
5
4
3
2
1
```

The status bar at the bottom right indicates the file is 3.13.9 (Microsoft Store) and the date is 17-02-2026.

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():
```

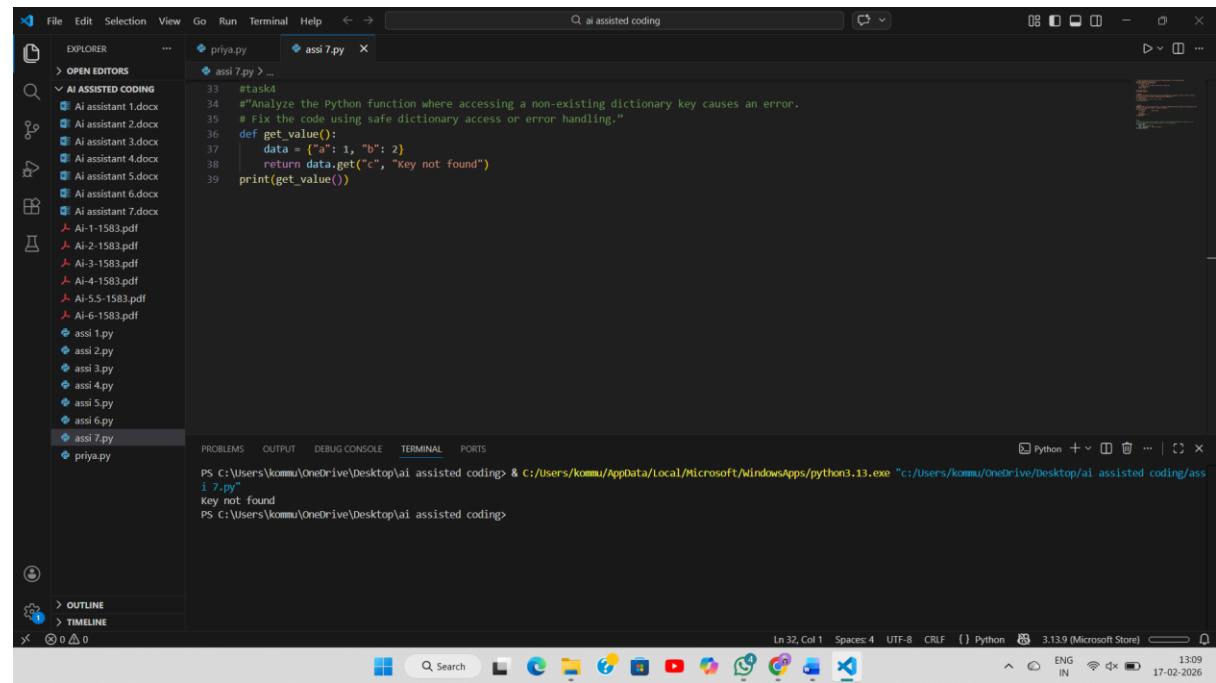
```
    data = {"a": 1, "b": 2}
```

```
    return data["c"]
```

```
print(get_value())
```

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

code:



The screenshot shows the Microsoft Visual Studio Code interface. The code editor displays the following Python script:

```
33 #task4
34 """Analyze the Python function where accessing a non-existing dictionary key causes an error.
35 # Fix the code using safe dictionary access or error handling."""
36 def get_value():
37     data = {"a": 1, "b": 2}
38     return data.get("c", "key not found")
39 print(get_value())
```

The terminal below the code editor shows the output of running the script:

```
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & c:/users/kommu/appData/local/Microsoft/WindowsApps/python3.13.exe "c:/Users/kommu/OneDrive/Desktop/ai assisted coding/ass1.py"
Key not found
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>
```

The status bar at the bottom right indicates the file is saved (green checkmark), the Python extension is active, the Python version is 3.13.9 (Microsoft Store), and the date and time are 17-02-2026 13:09.

Task 5 (Infinite Loop – Wrong Condition)

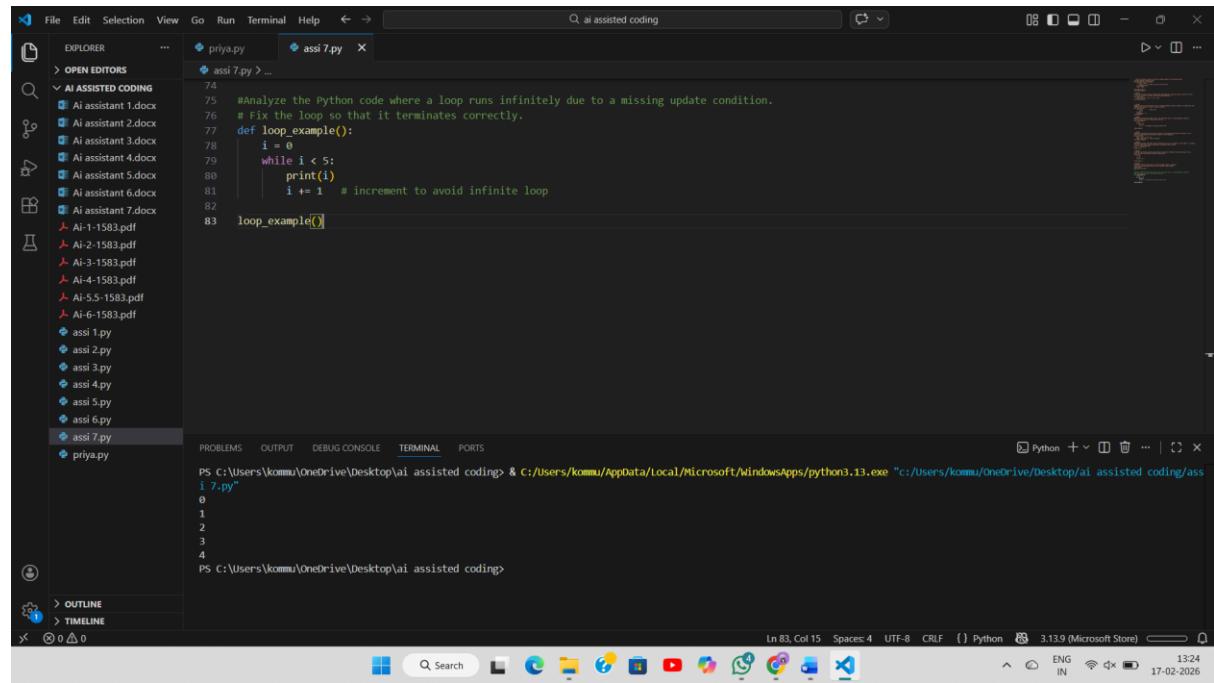
Task: Analyze given code where loop never ends. Use AI to detect and fix it.

Bug: Infinite loop

```
def loop_example():
    i = 0
    while i < 5:
        print(i)
```

Expected Output: Corrected loop increments i.

Code:



The screenshot shows the Microsoft Visual Studio Code interface. The code editor displays a Python script named `assi 7.py` with the following content:

```
74  #Analyze the Python code where a loop runs infinitely due to a missing update condition.
75  # fix the loop so that it terminates correctly.
76
77 def loop_example():
78     i = 0
79     while i < 5:
80         print(i)
81         i += 1  # increment to avoid infinite loop
82
83 loop_example()
```

The terminal below shows the output of running the script:

```
PS C:\Users\kommu\Desktop\ai assisted coding> & c:/users/kommu/appdata/local/microsoft/windowsapps/python3.11.exe "c:/users/kommu/Desktop/ai assisted coding/assi 7.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.

Code:

```
File Edit Selection View Go Run Terminal Help < > Q ai assisted coding
EXPLORER ... priya.py assi 7.py > ...
OPEN EDITORS
AI ASSISTED CODING
Ai assistant 1.docx
Ai assistant 2.docx
Ai assistant 3.docx
Ai assistant 4.docx
Ai assistant 5.docx
Ai assistant 6.docx
Ai assistant 7.docx
Ai-1-1583.pdf
Ai-2-1583.pdf
Ai-3-1583.pdf
Ai-4-1583.pdf
Ai-5-1583.pdf
Ai-6-1583.pdf
assi 1.py
assi 2.py
assi 3.py
assi 4.py
assi 5.py
assi 6.py
assi 7.py
priya.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & c:/users/kommu/appData/local/Microsoft/WindowsApps/python3.13.exe "c:/users/kommu/OneDrive/Desktop/ai assisted coding/assi 7.py"
1 2
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>
Ln 46, Col 12. Spaces: 4 UTF-8 CRLF {} Python 3.13.9 (Microsoft Store) 13:11 ENG IN 17-02-2026
```

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.

code:

The screenshot shows the Microsoft Visual Studio Code interface. The left sidebar displays the 'EXPLORER' view with various files and documents. The main editor area shows the 'ass1.py' file with the following code:

```
48     #task6
49     """Analyze the Python function where mixed or incorrect indentation causes execution errors.
50     # Fix the code by applying consistent indentation."""
51     def func():
52         x = 5
53         y = 10
54         return x + y
55
56     print(func())
```

The terminal at the bottom shows the command being run and its output:

```
PS C:\Users\kommu\Desktop\ai assisted coding> & c:/users/kommu/appdata/local/microsoft/windowsapps/python3.11.exe "c:/users/kommu/Desktop/ai assisted coding/ass1.py"
15
PS C:\Users\kommu\Desktop\ai assisted coding>
```

The status bar at the bottom right indicates the file is 3.13.9 (Microsoft Store) and the date is 17-02-2026.

Task 8 (Import Error – Wrong Module Usage)

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

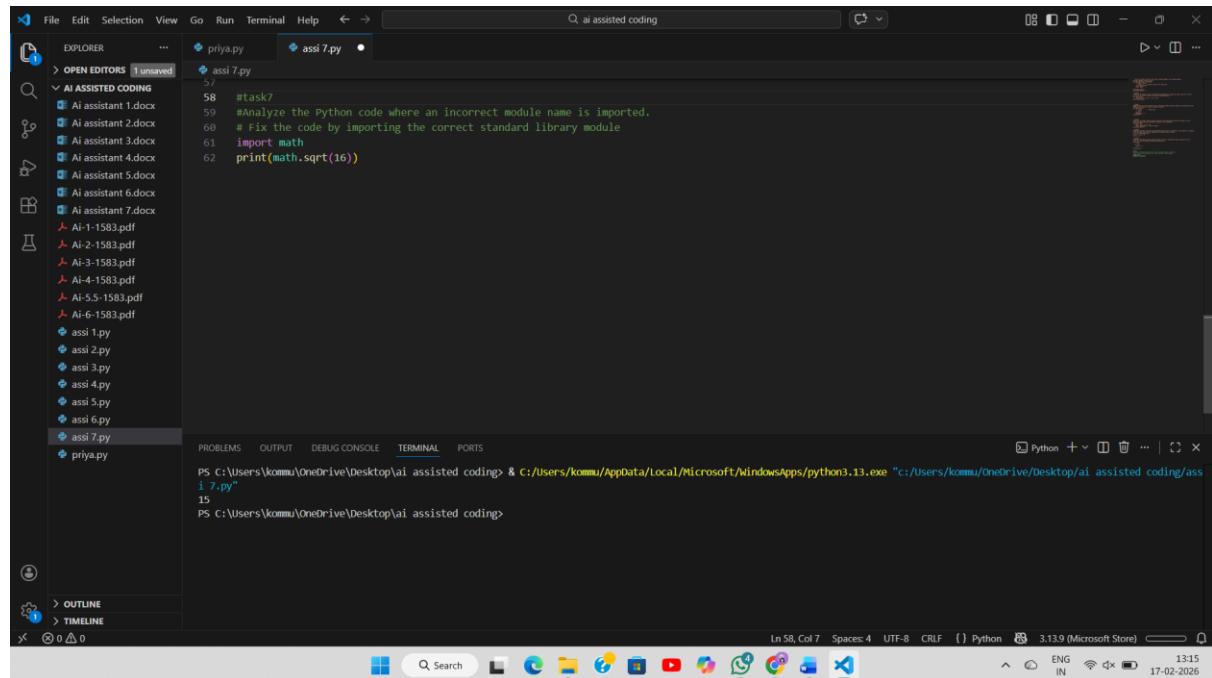
Bug: Wrong import

```
import maths
```

```
print(maths.sqrt(16))
```

Expected Output: Corrected to import math

Code:



```
58 #task7
59 #Analyze the Python code where an incorrect module name is imported.
60 # Fix the code by importing the correct standard library module
61 import maths
62 print(maths.sqrt(16))

PS C:\Users\kommu\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/kommu/Desktop/ai assisted coding/assi 7.py"
15
PS C:\Users\kommu\Desktop\ai assisted coding>
```