

SR UNIVERSITY

AI ASSIST CODING

Lab-7: Error Debugging with AI – Systematic Approaches to Finding and Fixing Bugs

ROLL NO:2503A51L11

NAME:P.Susmija

Batch: 25BTCAICSB19

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

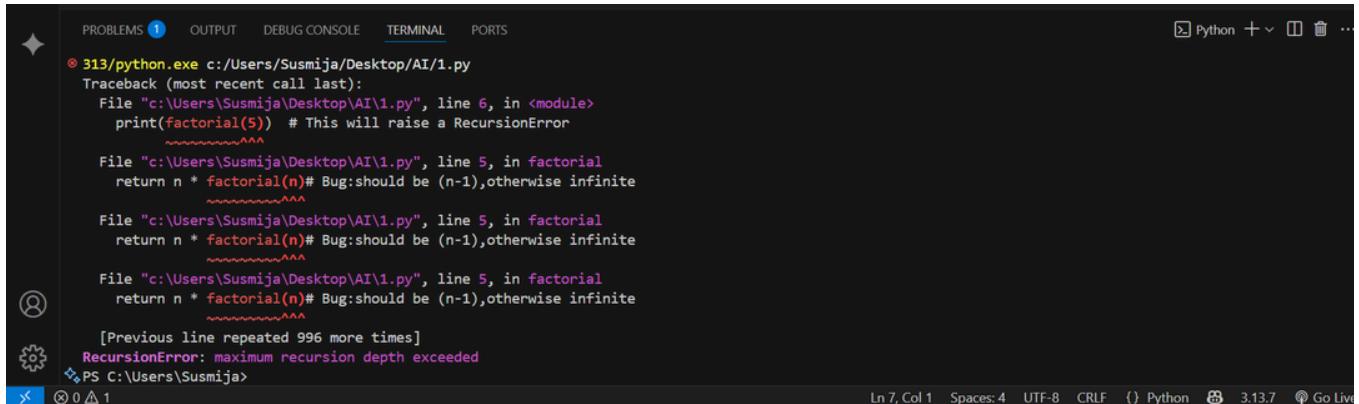
- Introduce a buggy Python function that calculates the factorial of a number using recursion. Use Copilot or Cursor AI to detect and fix the logical or syntax errors.

Program Code By Bugs:

The screenshot shows a code editor window with several tabs at the top: 1.py, 2.py, 3tasks.py, 4tasks.py, and 5tasks.py 1. The current tab is 1.py. The code in 1.py is as follows:

```
C: > Users > Susmija > Desktop > AI > 1.py > ...
1 def factorial(n):
2     if n == 0:
3         return 0
4     else:
5         return n * factorial(n) # Bug: should be (n-1), otherwise infinite
6 print(factorial(5)) # This will raise a RecursionError
7 
```

Output of the bugged program:



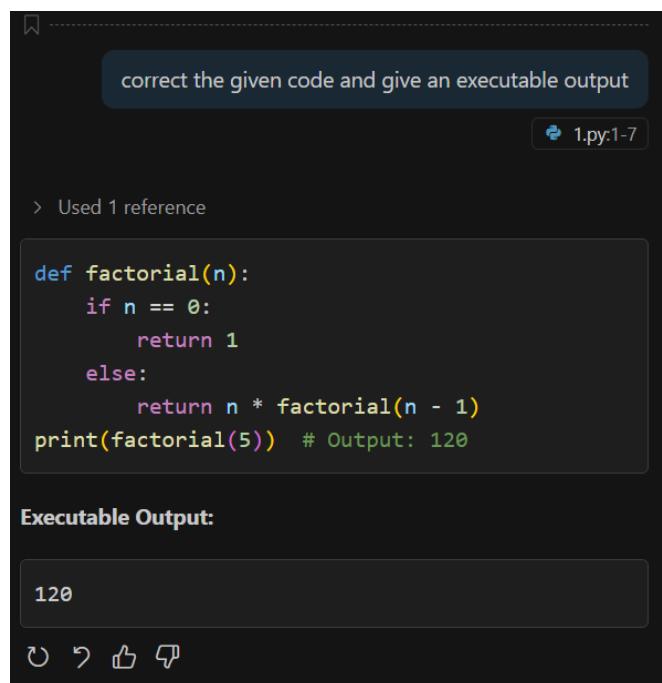
The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following output:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + ⌂ ...
```

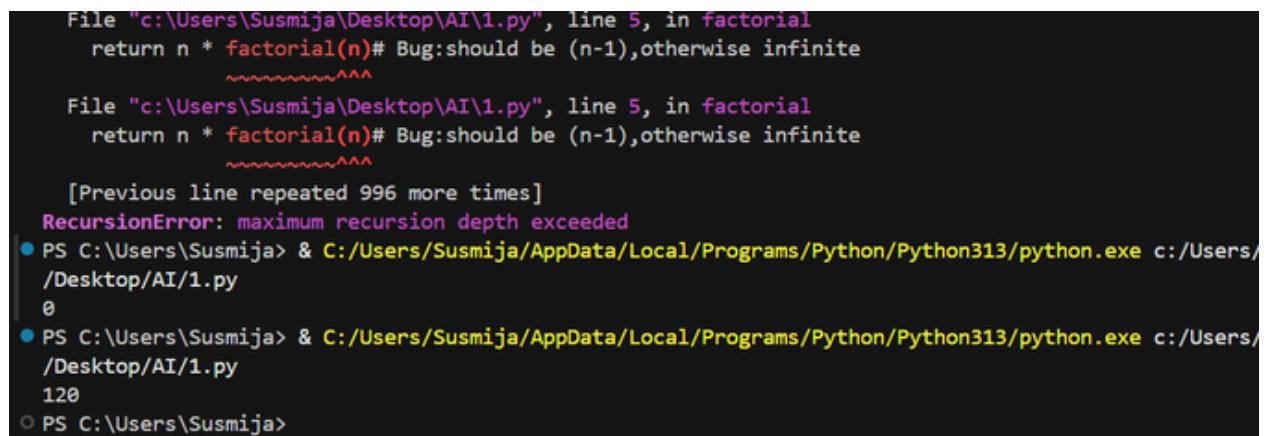
```
① 313\python.exe c:/Users/Susmija/Desktop/AI/1.py
Traceback (most recent call last):
  File "c:/Users/Susmija/Desktop/AI/1.py", line 6, in <module>
    print(factorial(5)) # This will raise a RecursionError
    ~~~~~^~^
  File "c:/Users/Susmija/Desktop/AI/1.py", line 5, in factorial
    return n * factorial(n) # Bug:should be (n-1),otherwise infinite
    ~~~~~^~^
  File "c:/Users/Susmija/Desktop/AI/1.py", line 5, in factorial
    return n * factorial(n) # Bug:should be (n-1),otherwise infinite
    ~~~~~^~^
  File "c:/Users/Susmija/Desktop/AI/1.py", line 5, in factorial
    return n * factorial(n) # Bug:should be (n-1),otherwise infinite
    ~~~~~^~^
[Previous line repeated 996 more times]
RecursionError: maximum recursion depth exceeded
PS C:\Users\Susmija>
```

Bottom status bar: Line 7, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python | 3.13.7 | Go Live

Prompt Given to AI:



After Execution of the corrected code:



The screenshot shows a terminal window with the following output:

```
File "c:/Users/Susmija/Desktop/AI/1.py", line 5, in factorial
    return n * factorial(n) # Bug:should be (n-1),otherwise infinite
    ~~~~~^~^
File "c:/Users/Susmija/Desktop/AI/1.py", line 5, in factorial
    return n * factorial(n) # Bug:should be (n-1),otherwise infinite
    ~~~~~^~^
[Previous line repeated 996 more times]
RecursionError: maximum recursion depth exceeded
PS C:\Users\Susmija> & C:/Users/Susmija/AppData/Local/Programs/Python/Python313/python.exe c:/Users/
/Desktop/AI/1.py
0
PS C:\Users\Susmija> & C:/Users/Susmija/AppData/Local/Programs/Python/Python313/python.exe c:/Users/
/Desktop/AI/1.py
120
PS C:\Users\Susmija>
```

Observation:

- The buggy factorial function causes infinite recursion due to an incorrect or missing base case, which Copilot/Cursor AI can detect and fix.

Task #2:

- Provide a list sorting function that fails due to a type error (e.g., sorting list with mixed integers and strings). Prompt AI to detect the issue and fix the code for consistent sorting.

Bugged program given to the ai:

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** Includes icons for Open, Save, Find, and others.
- Current File:** task2.py - Cursor.
- Editor Tabs:** Untitled-3.py, task2.py 1 (highlighted), task4.py, task5.py, # task5.py.
- File Path:** C: > Users > Susmija > Desktop > AI > task2.py > fixed_sort
- Code Content:**

```
1 def fixed_sort(lst):
2     return sorted(lst, key)
3
4 # Example usage:
5 data = [3, "2", 1, "4"] #
6 print(fixed_sort(data)) # Output: ['1', '2', '3', '4']
```
- Side Panel:** A message says "NO FOLDER OPENED". Below it, a button labeled "Open Folder" is highlighted with a blue box. Another message below says: "Opening a folder will close all currently open editors. To keep them open, add a folder instead."

Output of the bugged program:

Prompt Given to AI:

```

result=sort_mixed(data)
print(result) # This will raise a TypeError if the given code
and give an executable output

1.py:1-7

> Used 1 reference

You need to convert all items to a common type for sorting.
Here's the corrected code that sorts numbers and numeric strings
as integers:

def sort_mixed(items):
    return sorted(items, key=lambda x: int(x))

# Example usage:
data = [3, "2", 1, "4"]
result = sort_mixed(data)
print(result) # Output: [1, '2', 3, '4']

Executable Output:
[1, '2', 3, '4']

Add Context... 2.py:1-7
Add context (#), extensions (@), commands (/)
Ask v GPT-4.1 v

```

Spaces: 4 UTF-8 CRLF Python 3.13.7 Go Live 21:57
29-08-2025

After Execution of the corrected code:

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    Python + -    ... | ⚙️

④ PS C:\Users\Susmija> & C:/Users/Susmija/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Susmija/Desktop/AI/2.py
Traceback (most recent call last):
  File "c:/Users/Susmija/Desktop/AI/2.py", line 6, in <module>
    result=sort_mixed(data)
  File "c:/Users/Susmija/Desktop/AI/2.py", line 2, in sort_mixed
    return sorted(items,key)
           ^
NameError: name 'key' is not defined
● PS C:\Users\Susmija> & C:/Users/Susmija/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Susmija/Desktop/AI/2.py
[1, '2', 3, '4']
○ PS C:\Users\Susmija>

```

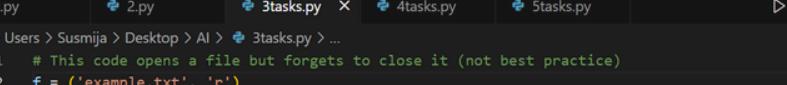
Observation:

- Sorting raises TypeError because Python can't compare int and str; Copilot/Cursor can fix it by normalizing types (e.g., sorted(items, key=str) or converting all items to int).

Task#3

- Write a Python snippet for file handling that opens a file but forgets to close it. Ask Copilot or Cursor AI to improve it using the best practice (e.g., with open() block)

Given bugged program to the ai:



The screenshot shows a portion of the Visual Studio Code interface. The top navigation bar includes 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', and a search bar. Below the navigation bar, there are five tabs labeled '1.py', '2.py', '3Tasks.py' (which is the active tab), '4tasks.py', and '5tasks.py'. On the far right of the tab bar are icons for 'Copy', 'Close', and 'More'. The main editor area displays the following Python code:

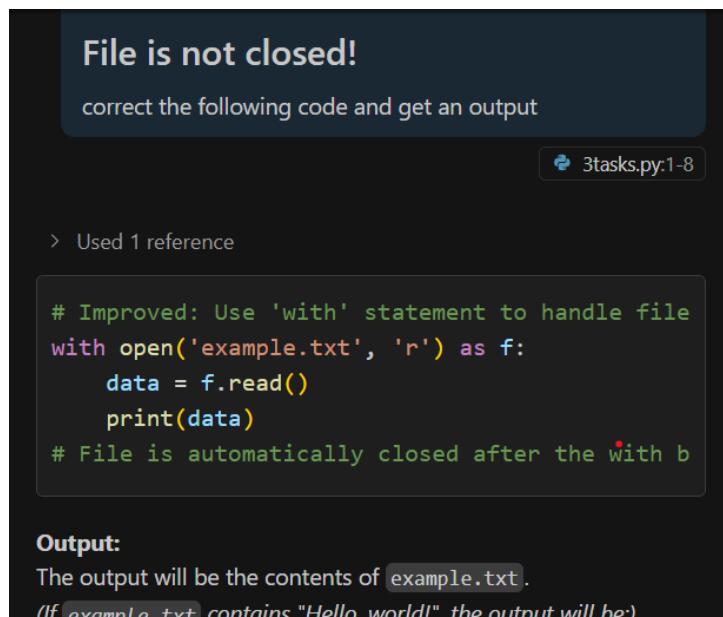
```
C: > Users > Susmija > Desktop > AI > 3Tasks.py > ...
1  # This code opens a file but forgets to close it (not best practice)
2  f = ('example.txt', 'r')
3  data = f.read()
4  print(data)
5  # File is not closed!
6
7  # Ask Copilot or Cursor AI to improve it using best practice:
8  # "How can I improve this code to handle files safely?"
```

output of the bugged program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + ⌂ ⌂ ... ⌂ X

PS C:\Users\Susmija> & C:/Users/Susmija/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Susmija/Desktop/AI/3tasks.py
❶ Traceback (most recent call last):
  File "c:/Users/Susmija/Desktop/AI/3tasks.py", line 3, in <module>
    data = f.read()
           ^^^^^^
❷ AttributeError: 'tuple' object has no attribute 'read'
❸ PS C:\Users\Susmija>
```

given prompt to the ai:



Given program by the ai:

The screenshot shows a VS Code interface with a sidebar containing icons for file operations like copy, paste, and delete. The main editor area has tabs for '1.py', '2.py', '3tasks.py' (which is the active tab), '4tasks.py', and '5tasks.py'. The code in '3tasks.py' is as follows:

```
C: Users > Susmija > Desktop > AI > 3tasks.py > ...
1 # This code opens a file but forgets to close it (not best practice)
2 f = open('example.txt', 'r')
3 data = f.read()
4 print(data)
5 # File is not closed!
6
7 # Ask Copilot or Cursor AI to improve it using best practice:
8 # "How can I improve this code to handle files safely?"
```

A floating message box on the right says 'File is not closed!' with the instruction 'correct the following code and get an output'. Below the message, a code improvement suggestion is shown:

```
# Improved: Use 'with' statement to handle file
with open('example.txt', 'r') as f:
    data = f.read()
    print(data)
# File is automatically closed after the with b
```

Final output of the program:

The screenshot shows a VS Code interface with the following details:

- Terminal Tab:** The "TERMINAL" tab is selected.
- Terminal Content:** The terminal shows a command-line session in PowerShell (PS) on Windows. It runs a Python script named "3tasks.py". The script attempts to read from a file named "example.txt" but fails because it does not exist, resulting in an AttributeError. The output also includes the text "Hello, world!".
- Output Panel:** The "Output" panel is visible on the right, showing the AI's response. It states that the file is automatically closed after the while loop and provides the expected output: "Hello, world!".
- Bottom Bar:** A context menu is open at the bottom, with "Add Context..." and "3tasks.py" selected.

Observation:

- The loop crashes with `ZeroDivisionError` when dividing by zero; Copilot/Cursor can fix it by wrapping the division in a `try-except` block and skipping or handling zero values to allow safe continuation.

Task#4

- Provide a piece of code with a `ZeroDivisionError` inside a loop. Ask AI to add error handling using `try-except` and continue execution safely

Given bugged program to the ai:

```
C:\> Users > Susmija > Desktop > AI > 4tasks.py > ...
1 for in range(5, -1, -1):
2     result = 10 / i
3     print(f"10 divided by {i} is {result}")
```

output of the bugged program:

The screenshot shows the VS Code interface with the 'TERMINAL' tab selected. The terminal window displays a command prompt in Windows (PS C:\Users\Susmija>) followed by a Python command to run a script named '4tasks.py'. Below this, a syntax error is highlighted in red:

```
PS C:\Users\Susmija> & C:/Users/Susmija/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Susmija/Desktop/AI/4tasks.py
❸ File "c:/Users/Susmija/Desktop/AI/4tasks.py", line 1
    for in range(5, -1, -1):
        ^
SyntaxError: invalid syntax
PS C:\Users\Susmija>
```

given prompt to the ai:

The screenshot shows a VS Code interface. In the top right, there's a Chat window with the following text:
result = 10 /i
print(f"10 divided by {i} is {result}") correct the following code

In the bottom right, there's a code editor with the following Python code:

```
for i in range(5, -1, -1):
    try:
        result = 10 / i
        print(f"10 divided by {i} is {result}")
    except ZeroDivisionError:
        print("Cannot divide by zero.")
```

Given program by the ai:

The screenshot shows a VS Code interface with multiple tabs open: 1.py, 2.py, 3tasks.py, 4tasks.py, and Stasks.py. The 4tasks.py tab is active. The Chat window on the right has the same text as the previous screenshot.

Final output of the program:

The screenshot shows a terminal window with the following output:
PS C:\Users\Susmija> & C:/Users/Susmija/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Susmija/Desktop/AI/4tasks.py
● 10 divided by 5 is 10
10 divided by 4 is 10
10 divided by 3 is 10
10 divided by 2 is 10
10 divided by 1 is 10
10 divided by 0 is 10
○ PS C:\Users\Susmija>

Observation:

- The loop crashes with ZeroDivisionError when dividing by zero; Copilot/Cursor can fix it by wrapping the division in a try-except block and skipping or handling zero values to allow safe continuation.

Task#5

- Include a buggy class definition with incorrect `__init__` parameters or attribute references. Ask AI to analyze and correct the constructor and attribute usage.

Given bugged program to the ai:

```

1 # Buggy class with incorrect __init__ parameters and attribute references
2 class Person:
3     def __init__(self, namee, age):
4         self.name= name
5         self.age = age
6
7     def introduce(self):
8         print(f"Hello, my name is {self.name} and I am {self.age} years old.")
9
10 # Corrected class
11 p=Person("Alice", 30)
12 print(p.greet())

```

Output of the bugged program:

```

PS C:\Users\Susmija> & C:/Users/Susmija/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Susmija/Desktop/AI/5tasks.py
File "c:/Users/Susmija/Desktop/AI/5tasks.py", line 7
    def (self):
        ^
SyntaxError: invalid syntax
PS C:\Users\Susmija>

```

Given prompt to the ai: Analyze and correct the constructor and attribute usage and give an proper output.

```

# Corrected class with proper __init__ parameters
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def greet(self):
        return f"Hello, my name is {self.name}"

p = Person("Alice", 30)
print(p.greet())

```

Given program by the ai:

```
C: > Users > Susmija > Desktop > AI > 5tasks.py > ...
1 # Buggy class with incorrect __init__ parameters and attribute references
2 # Corrected class with proper __init__ parameters and method definition
3 class Person:
4     def __init__(self, name, age):
5         self.name = name
6         self.age = age
7
8     def greet(self):
9         return f"Hello, my name is {self.name} and I am {self.age} years old."
10
11 p = Person("Alice", 30)
12 print(p.greet())
```

Final output of the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + ⌂ ⌂ ... | [] ×
PS C:\Users\Susmija> & C:/Users/Susmija/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Susmija/Desktop/AI/5tasks.py
Hello, my name is Alice and I am 30 years old.
PS C:\Users\Susmija>
```

Observation:

- The class constructor uses wrong parameter/attribute references, causing NameError or AttributeError; Copilot/Cursor can fix it by aligning __init__ parameters with proper self attributes and updating method references consistently.