

GIRUGULA VARSHINI

2403A51L14

Batch : 51

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

Task 1 – Runtime Error Due to Invalid Input Type

Prompt:

Write a Python function to determine whether a given number is prime.

Code :

The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for 'Welcome', 'day.py', 'AI.py', and '7.2.py'. The '7.2.py' tab is active. Below the tabs, the code is displayed:

```
7.2.py > ...
1 num = input("Enter a number: ")
2 result = num + 10 num = '5'
```

A red arrow points to the line 'result = num + 10 num = '5''. A tooltip or status bar at the bottom of the code area displays the error message:

Exception has occurred: TypeError
can only concatenate str (not "int") to str
File "C:\Users\girug\Downloads\AI\7.2.py", line 2, in <module>
 result = num + 10
 ^~~~~~
TypeError: can only concatenate str (not "int") to str

The code editor has a sidebar on the left labeled 'Variables'.

Output:

The screenshot shows the VS Code interface. At the top, there are tabs for 'Welcome', 'day.py', 'AI.py', and '7.2.py'. The '7.2.py' tab is active, displaying the following code:

```
1 num = int(input("Enter a number: "))
2 result = num + 10
3 print(result)
4
```

Below the code editor is a terminal window. The terminal tab is selected, showing the following command-line interaction:

```
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c:;
cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '--' 'c:\Users\girug\Downloads\AI\7.2.py'
Enter a number: 6
16
PS C:\Users\girug\Downloads\AI>
```

Justification: The program failed because `input()` returns data as a string, and adding a string to an integer is invalid in Python. The AI corrected this by converting the input to an integer using `int()`, ensuring the arithmetic operation works properly. This type conversion is necessary to match the expected numeric behavior of the program.

Task 2 – Incorrect Function Return Value

Prompt:

Generate a function to calculate the sum of elements in a list.

Code :

```
23
24
25
26
27
28 def square(n):
29     result = n * n
30
31
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\girug\Downloads\AI> c::; cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14.64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '51096' '--' 'C:\Users\girug\Downloads\AI\7.2.py'
PS C:\Users\girug\Downloads\AI> 6
6
PS C:\Users\girug\Downloads\AI> []
```

Output:

```
24
25
26
27 def square(n):
28     result = n * n
29     return result
30 print(square(8))
31

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

> c::; cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '638' '--' 'C:\Users\girug\Downloads\AI\7.2.py'
25
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c::; cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '639' '--' 'C:\Users\girug\Downloads\AI\7.2.py'
PS C:\Users\girug\Downloads\AI> 8^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c::; cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '502' '--' 'C:\Users\girug\Downloads\AI\7.2.py'
64
PS C:\Users\girug\Downloads\AI> []
```

In 30 Col 15

Justification:

Although the function computed the square internally, it never returned the result, causing the output to be lost. The AI identified the missing return statement and added it, allowing the function to properly

send the computed value back to the caller. Returning values is essential for functional correctness and reusability.

Task 3 – IndexError in List Traversal

Write a Python function that takes an alphanumeric string and returns only the digits.

Code :

The screenshot shows a code editor in VS Code with a dark theme. A Python file is open, showing lines 24 through 34. Line 32 contains a for loop that iterates from 0 to len(numbers)+1. Line 33 prints the element at index i. A tooltip above line 33 indicates that numbers = [10, 20, 30]. A red arrow points to the '+1' in the range function. A callout box highlights the error message: 'Exception has occurred: IndexError × list index out of range'. Below it, the stack trace shows the file 'C:\Users\girug\Downloads\AI\7.2.py', line 33, in <module>. The error message 'IndexError: list index out of range' is also visible. The bottom of the screen shows the VS Code interface with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. A status bar at the bottom shows 'x ou...' and '33:16'.

Output:

The screenshot shows a terminal window in VS Code with a dark theme. It displays the following session:

```
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54 numbers = [10, 20, 30]
55 for i in range(0, len(numbers)):
56 |   print(numbers[i])
57
```

Below the code, the terminal shows the command PS C:\Users\girug\Downloads\AI> followed by the execution of the script. The output shows the numbers 10, 20, and 30 printed sequentially. The bottom right corner of the terminal shows 'Ln 52, Col 1 Spaces: 4 UTF-8 CR'.

Justification:

The loop incorrectly iterated one step beyond the valid index range using `len(numbers) + 1`, causing an `IndexError`. AI fixed the boundary to `range(len(numbers))`, ensuring safe access of all existing list elements. This correction is justified because valid indices only go from 0 to `len(numbers)-1`.

Task 4 – Uninitialized Variable Usage

Prompt :

Write a Python function to count the number of vowels in a given string.

Code :

A screenshot of a code editor showing a Python script named 7.2.py. The code contains a single line: `print(total)`. A yellow box highlights this line, and a tooltip displays the error message: "Exception has occurred: NameError × name 'total' is not defined". Below the code editor, the terminal output shows the same error message: "NameError: name 'total' is not defined".

```
54
55
56
57 if True:
58 | pass
D 59 print(D total)

Exception has occurred: NameError ×
name 'total' is not defined

File "c:\Users\girug\Downloads\AI\7.2.py", line 59, in <module>
    print(total)
           ^^^^^^

NameError: name 'total' is not defined

'total' ...
59:7
60
61
62
```

Output:

A screenshot of a code editor showing the same Python script 7.2.py. This time, the code includes an initialization line: `total = 0 # Initializing the variable`. The terminal output shows the command `python 7.2.py` being run, and the output is `0`, indicating the program ran successfully. The status bar at the bottom right shows "Ln 79, Col 1 Spaces: 4 UTF-8".

```
77
78
79
80 total = 0 # Initializing the variable
81 if True:
82 | pass
83
84 print(total)
85

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

'--> 'C:\Users\girug\Downloads\AI\7.2.py'
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c:; cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\vscode\extensions\ms-python.python-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '63589'
'--> 'C:\Users\girug\Downloads\AI\7.2.py'
0
PS C:\Users\girug\Downloads\AI>
```

Justification:

The program attempted to print a variable (total) before it had been assigned any value, resulting in a runtime error. AI resolved this by initializing the variable to 0 before use, ensuring the program has a valid reference. Proper initialization prevents undefined behavior and is a fundamental programming requirement.

Task 5 – Logical Error in Student Grading System

Prompt :

write a Python function that takes three numbers and returns the minimum value without using min().

Code :

The screenshot shows the VS Code interface with the terminal tab selected. The code editor contains the following Python script:

```
79     marks = 85
80     if marks >= 90:
81         grade = "A"
82     elif marks >= 80:
83         grade = "C"
84     else:
85         grade = "B"
86     print(grade)
87
88
89
90
91
92
```

The terminal window shows the following output:

```
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c:;
cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '50305'
'--' 'C:\Users\girug\Downloads\AI\7.2.py'
C
PS C:\Users\girug\Downloads\AI>
```

Output:

The screenshot shows the VS Code interface with the terminal tab selected. The code editor contains the same Python script as before. The terminal window shows the following output:

```
74
75
76
77
78
79     marks = 85
80
81     if marks >= 90:
82         grade = "A"
83     elif marks >= 80:
84         grade = "B"
85     else:
86         grade = "C"
87
88     print(grade)
89
```

The terminal window shows the following output:

```
C
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI> c;;
cd 'c:\Users\girug\Downloads\AI'; & 'c:\Users\girug\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\girug\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '64291'
'--' 'C:\Users\girug\Downloads\AI\7.2.py'
B
PS C:\Users\girug\Downloads\AI>
```

Justification:

The conditions for assigning grades were incorrectly ordered, making the program assign a wrong grade for certain mark ranges. AI fixed this by arranging the conditions in a logically descending order (A → B → C), ensuring accurate evaluation. Correct conditional structure is essential for producing correct program decisions.