

AI ASSISTANT CODING

ASSIGNMENT-7.2

Course Title : AI Assisted Coding

Name : Saini Kirthan

Batch No : 34

H.NO : 2303A52157

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

Task Description

Task 1 – Runtime Error Due to Invalid Input Type

- A Python program accepts user input and performs arithmetic operations. However, the program throws a runtime error because the input is treated as a string instead of a numeric type.

Example (Buggy Code):

```
num = input("Enter a number: ")  
result = num + 10  
print(result)
```

- **Task:**

Use AI tools to identify the cause of the runtime error and modify the program so it executes correctly.

The screenshot shows a Python code editor interface with two code snippets and their corresponding terminal outputs.

Top Snippet:

```
Lab.Ass_7.2.py > ...
1  """Prompt - "Identify the runtime error in the following Python program
2  where user input is used in an arithmetic operation. Explain why the
3  error occurs and modify the code so that the program executes correctly
4  without errors."""
5
6  # Buggy code:
7  num = input("Enter a number: ")
8  res = num + 10
9  print(res)
```

Terminal Output (Top):

```
PS C:\Users\hp\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/hp/OneDrive/Desktop/AI ASSISTANT CODING/Lab_Ass_7.2.py"
Enter a number: 10
Traceback (most recent call last):
  File "c:/Users/hp/OneDrive/Desktop/AI ASSISTANT CODING/Lab_Ass_7.2.py", line 8, in <module>
    res = num + 10
           ~~~~~~
TypeError: can only concatenate str (not "int") to str
PS C:\Users\hp\OneDrive\Desktop\AI ASSISTANT CODING>
```

Bottom Snippet:

```
Lab.Ass_7.2.py > ...
9  """Fixed code"""
10
11 # Take input from the user
12 # input() always returns the value as a string
13 num = input("Enter a number: ")
14
15 # Convert the string input to an integer
16 # This is required to perform arithmetic operations
17 num = int(num)
18
19 # Perform arithmetic operation (now both are integers)
20 result = num + 10
21
22 # Print the final result
23 print(result)
```

Terminal Output (Bottom):

```
PS C:\Users\hp\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/hp/OneDrive/Desktop/AI ASSISTANT CODING/Lab_Ass_7.2.py"
Enter a number: 10
120
PS C:\Users\hp\OneDrive\Desktop\AI ASSISTANT CODING>
```

Task Description:

Task 2 – Incorrect Function Return Value

A function is designed to calculate the square of a number, but it does not return the computed result properly.

Example (Buggy Code):

```
def square(n):
    result = n * n
```

Task:

Use AI assistance to analyze the function and ensure the correct value is returned

The screenshot shows a code editor interface with two tabs: 'Task_2.py' and 'Lab_Ass_7.2.py'. The 'Task_2.py' tab contains the following code:

```
1 """  
2 Prompt - Analyze the following Python function that is intended to calculate the square of a number  
3 but does not return the result correctly. Identify the issue in the function and  
4 modify it so that it returns the correct output  
5 """  
6 # Buggy code:  
7 def square(n):  
8     # This line correctly calculates the square of the number  
9     result = n * n  
10    # There is no return statement in this function  
11    # Without 'return', Python automatically returns None  
12    print(square(3)) # This will print 'None' instead of '9'  
13  
14
```

The terminal below shows the output of running the code: 'None'.

The 'Lab_Ass_7.2.py' tab contains the following code:

```
15 """Fixed code"""  
16 def square(n):  
17     # Calculate the square of the given number  
18     result = n * n  
19  
20     # Return the calculated value to the caller  
21     # This ensures the function outputs the correct result  
22     return result  
23 print(square(3))
```

The terminal below shows the output of running the corrected code: '9'.

Task Description

Task 3 – Index Error in List Traversal

A Python program iterates over a list using incorrect index limits, causing an `IndexError`.

Example (Buggy Code):

```
numbers = [10, 20, 30]  
  
for i in range(0, len(numbers)+1):  
    print(numbers[i])
```

Task:

Use AI to identify the incorrect loop boundary and correct the iteration logic.

```
Task_3.py > ...
1  """
2  Prompt - Analyze the following Python program that raises an IndexError while iterating over a list.
3  Identify the incorrect loop boundary, explain why the error occurs, and modify the code to prevent
4  out-of-range list access."""
5  # Buggy code:
6  numbers = [10, 20, 30]
7
8  # Loop goes one step beyond the valid index range
9  for i in range(0, len(numbers)+1):
10     print(numbers[i])
11
12  """Fixed code"""
13  numbers = [10, 20, 30]
14
15  # Loop runs from index 0 to len(numbers) - 1
16  for i in range(0, len(numbers)):
17      print(numbers[i])
```

Task Description

Task 4 – Uninitialized Variable Usage

A program uses a variable in a calculation before assigning it any value.

Example (Buggy Code):

```
if True:
    pass
    print(total)
```

Task:

Use AI tools to detect the uninitialized variable and correct the program.

Task Description

Task 5 – Logical Error in Student Grading System

A grading program assigns incorrect grades due to improper conditional logic.

Example (Buggy Code):

marks = 85

```
if marks >= 90:
```

grade = "A"

elif marks >= 80:

grade = "C"

else:

grade = "B"

```
print(grade)
```

Task:

Use AI to analyze the grading conditions and correct the logical flow.

Task_5.py > ...

```
1 """
2 Prompt - Analyze the following Python grading program that assigns incorrect grades due to logical
3 errors in conditional statements. Identify the issue in the grading logic, explain why the output
4 is incorrect, and correct the conditional flow so that grades are assigned accurately.
5 """
6 # Buggy code:
7 marks = 85
8
9 if marks >= 90:
10     grade = "A"
11 elif marks >= 80:
12     grade = "C"
13 else:           ↴ B
14     grade = "B"
15
16 print(grade)
17
```

PROBLEMS DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\hp\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/hp/OneDrive/Desktop/AI ASSISTANT CODING/Task_5.py"
C

PS C:\Users\hp\OneDrive\Desktop\AI ASSISTANT CODING>

Task_5.py > ...

```
19 """Fixed code"""
20 marks = 85
21
22 # Check for highest grade first
23 if marks >= 90:
24     grade = "A"
25
26 # Check for the next lower grade range
27 elif marks >= 80:
28     grade = "B"
29
30 # Remaining marks fall into the lowest grade
31 else:
32     grade = "C"
33
34 print(grade)
```

PROBLEMS DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\hp\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/hp/OneDrive/Desktop/AI ASSISTANT CODING/Task_5.py"
B

PS C:\Users\hp\OneDrive\Desktop\AI ASSISTANT CODING>