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

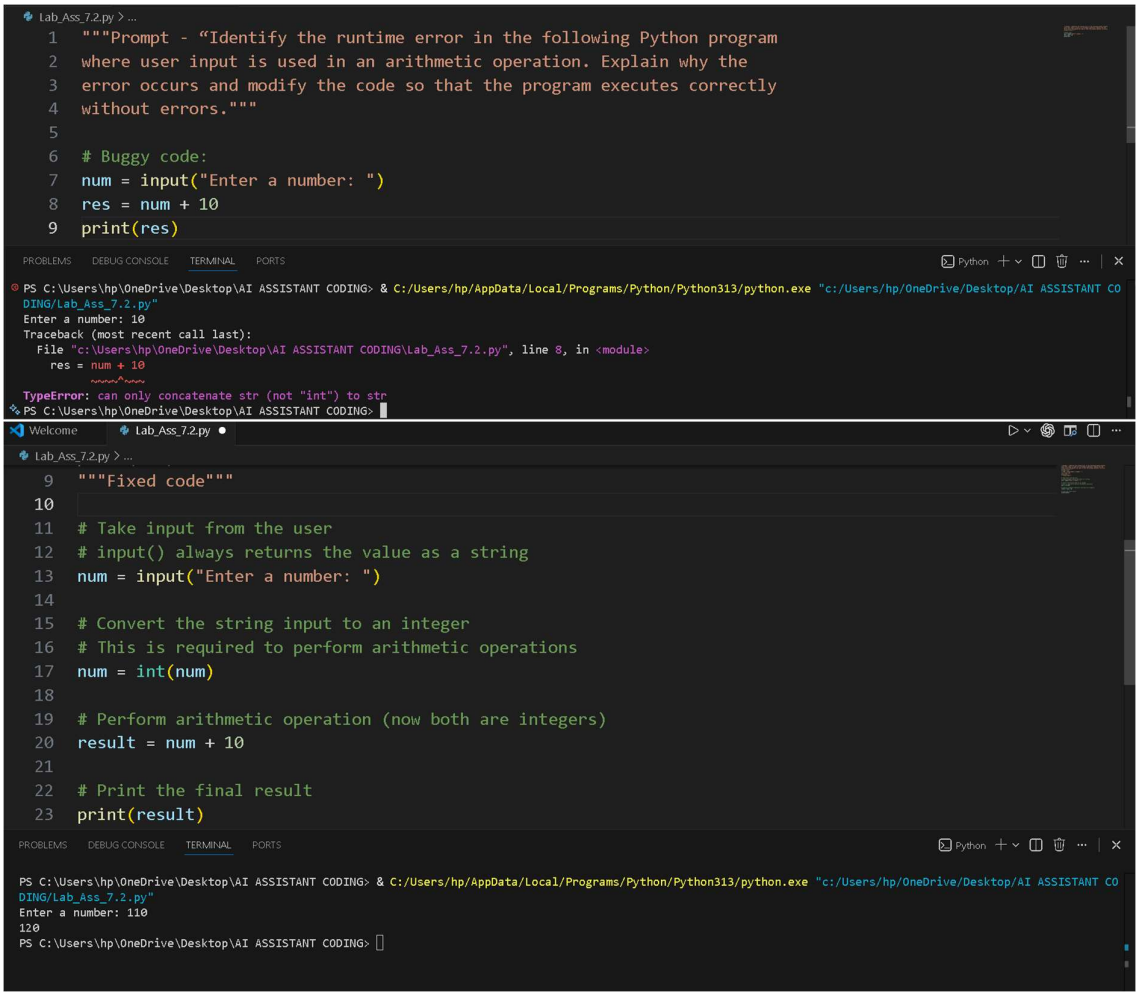
• A Python program accepts user input and performs arithmetic operations. However, the program throws a runtime error because the input is treated as a string instead of a numeric type.

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.



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.

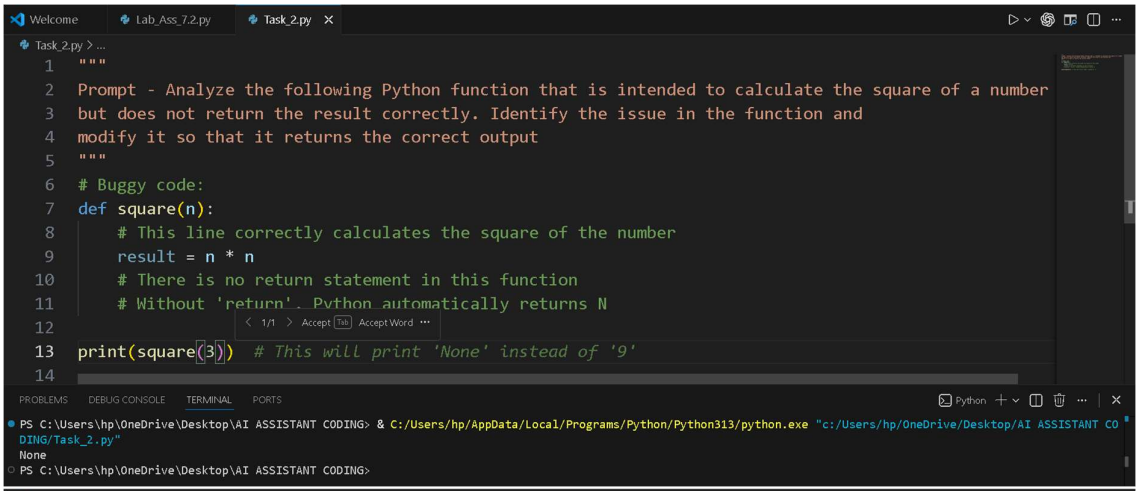
Example (Buggy Code):

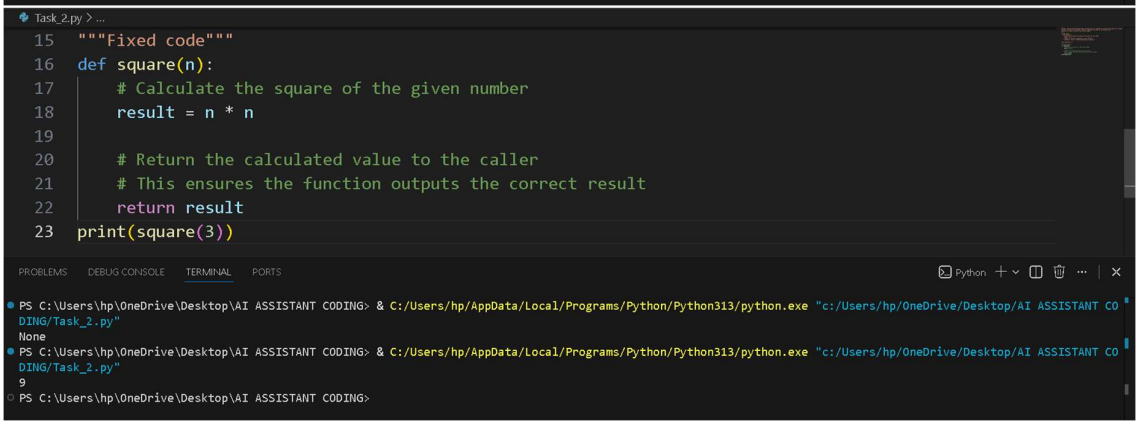
def square(n):

result = n * n

Task:

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





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]

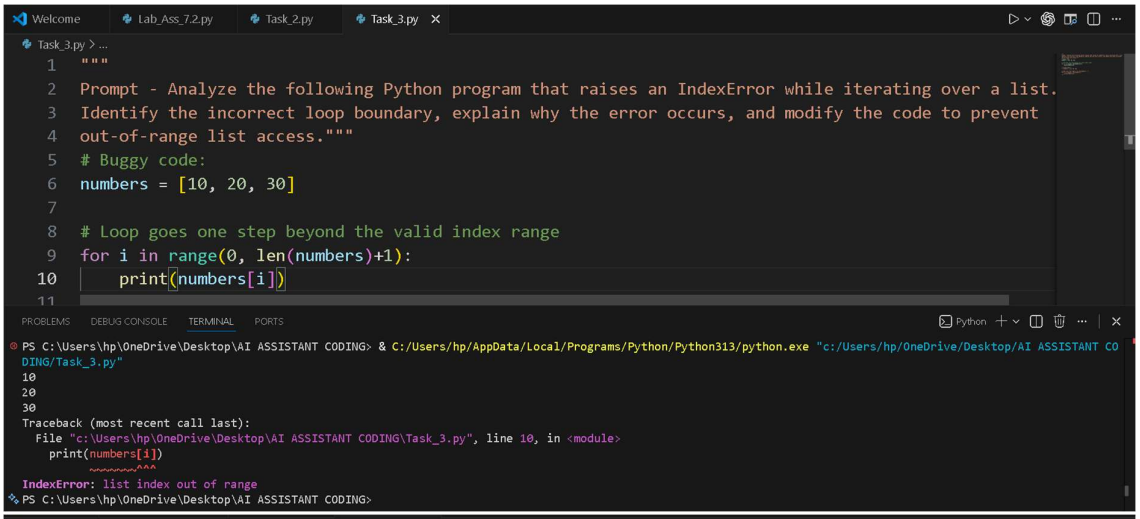
for i in range(0, len(numbers)+1):

print(numbers[i])

Task:

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

logic.



Task Description

Task 4 – Uninitialized Variable Usage

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

Example (Buggy Code):

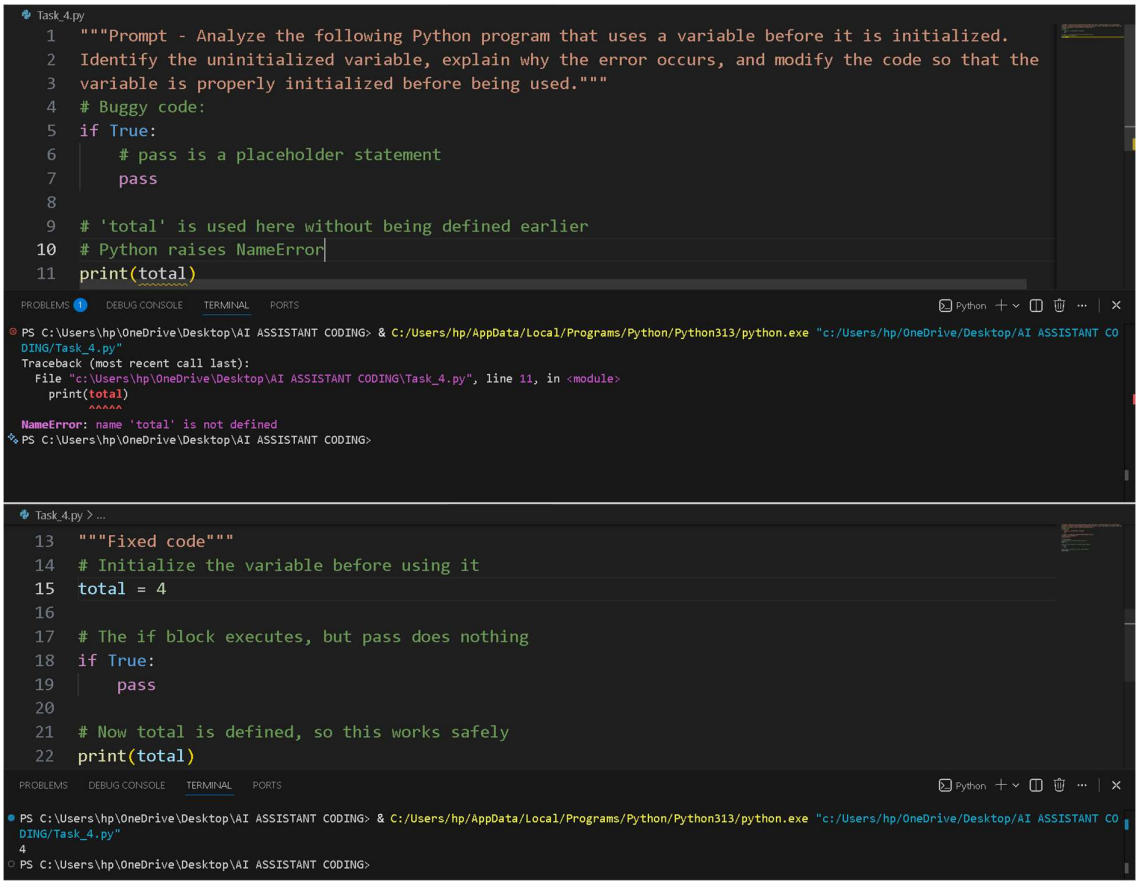
if True:

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"

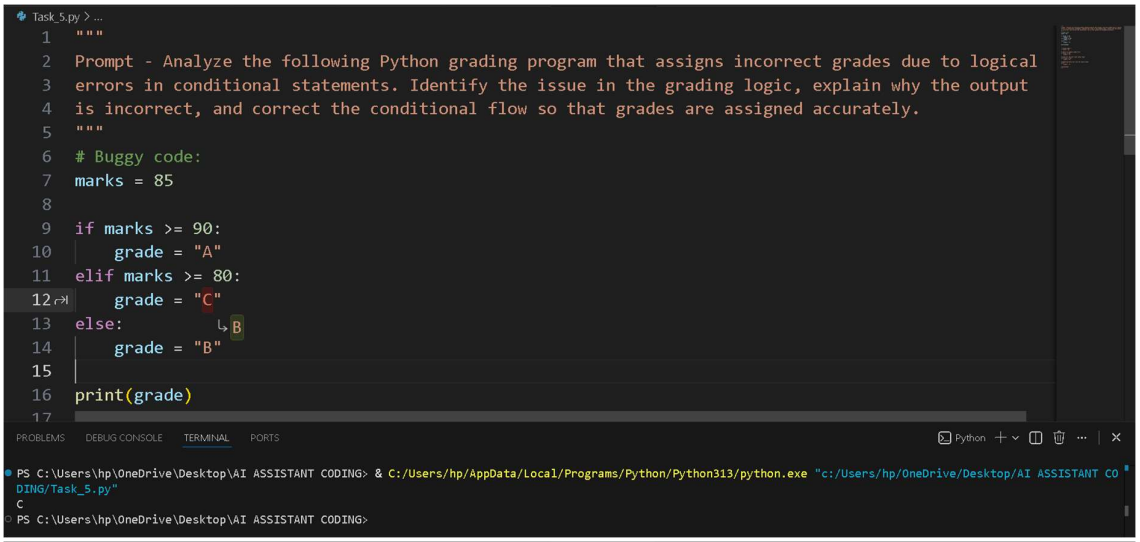
else:

grade = "B"

print(grade)

Task:

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

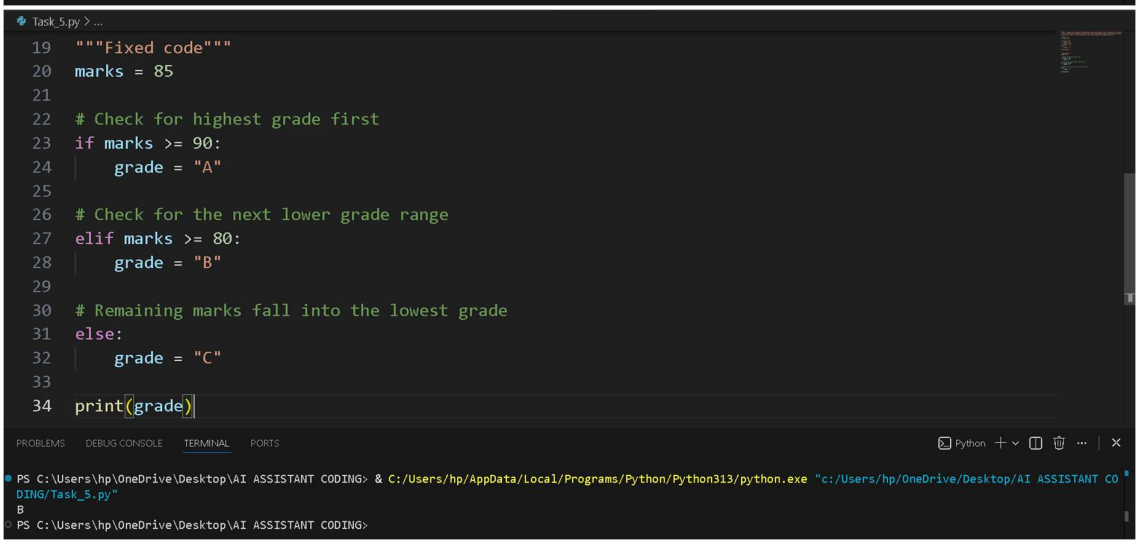


```
"""
Prompt - Analyze the following Python grading program that assigns incorrect grades due to logical
errors in conditional statements. Identify the issue in the grading logic, explain why the output
is incorrect, and correct the conditional flow so that grades are assigned accurately.
"""
# Buggy code:
marks = 85

if marks >= 90:
    grade = "A"
elif marks >= 80:
    grade = "C"
else:
    grade = "B"

print(grade)
```

```
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>
```



```
"""Fixed code"""
marks = 85

# Check for highest grade first
if marks >= 90:
    grade = "A"

# Check for the next lower grade range
elif marks >= 80:
    grade = "B"

# Remaining marks fall into the lowest grade
else:
    grade = "C"

print(grade)
```

```
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>
```