sru SR UNIVERSITY

School of Computer Science and Artificial Intelligence

Lab Assignment # 7.2

Program	: B. Tech (CSE)
Specialization	
Course Title	: AI Assisted Coding
Course Code	: 23CS002PC304
Semester	11
Academic Session	: 2025-2026
Name of Student	:.P.ESHWAR
Enrollment No.	: 2403A51L26
Batch No.	51
Date	: 30/01/26

Submission Starts here

Screenshots:

Task 1 — **Runtime Error Due to Invalid Input Type**

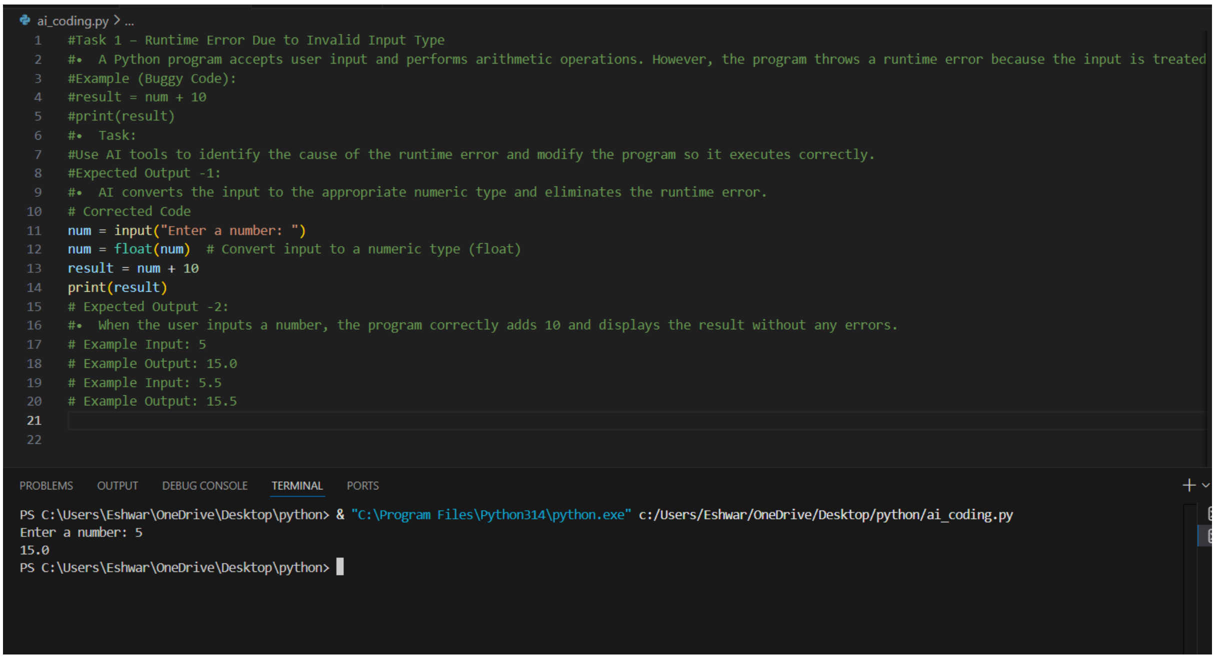
The issue is fixed by converting the input to a numeric type using float0 and handling invalid input with try-except

(Buggy Code):

```
num = input("Enter a number: ")

result = num + 10

print(result)
```



```python
#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
#Example (Buggy Code):
#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.
#Expected Output -1:
#•  AI converts the input to the appropriate numeric type and eliminates the runtime error.
# Corrected Code
num = input("Enter a number: ")
num = float(num)  # Convert input to a numeric type (float)
result = num + 10
print(result)
# Expected Output -2:
#•  When the user inputs a number, the program correctly adds 10 and displays the result without any errors.
# Example Input: 5
# Example Output: 15.0
# Example Input: 5.5
# Example Output: 15.5
```

```
PS C:\Users\Eshwar\OneDrive\Desktop\python> & "C:\Program Files\Python314\python.exe" c:/Users/Eshwar/OneDrive/Desktop/python/ai_coding.py
Enter a number: 5
15.0
PS C:\Users\Eshwar\OneDrive\Desktop\python> 
```

Task 2 - Incorrect Function Return Value

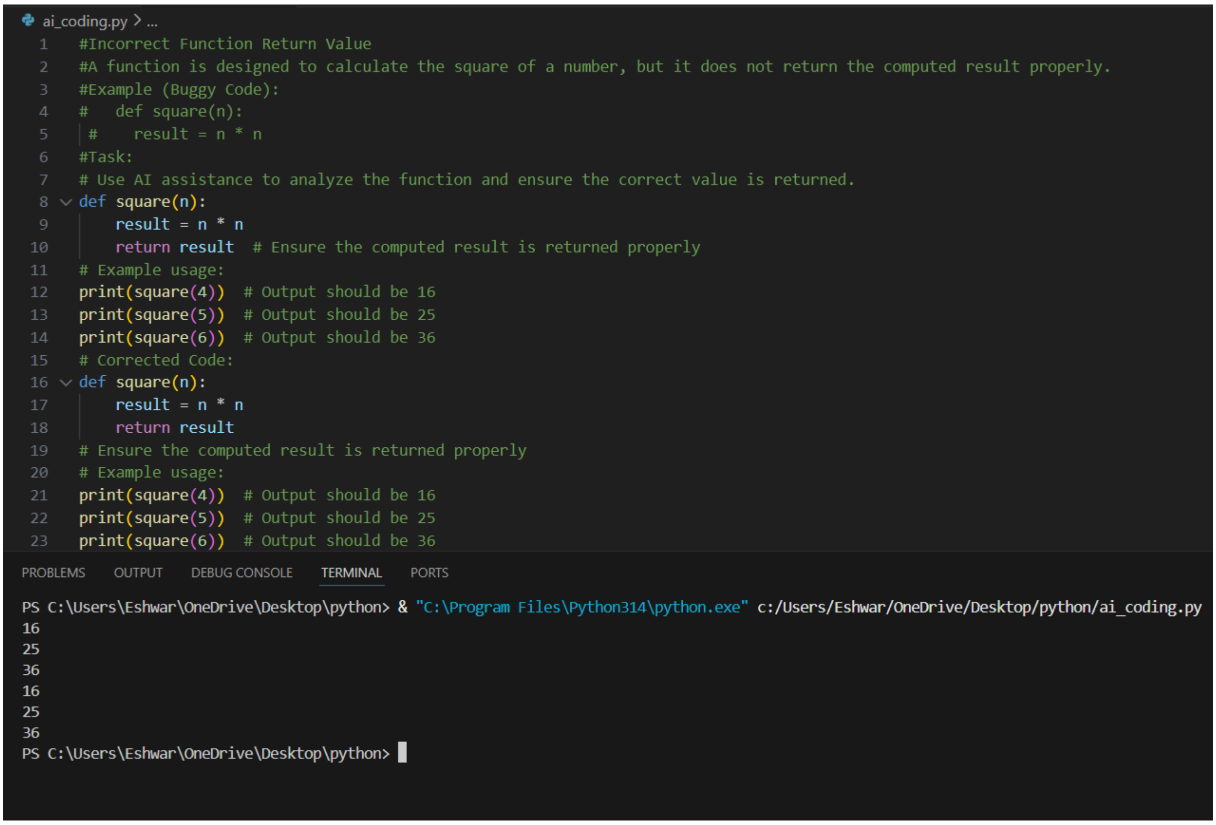
A function is designed to calculate the square of a number, but it does not return the computed result properly.

Example (Buggy Code):

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

Task:

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



```
#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.
def square(n):
    result = n * n
    return result  # Ensure the computed result is returned properly
# Example usage:
print(square(4))  # Output should be 16
print(square(5))  # Output should be 25
print(square(6))  # Output should be 36
# Corrected Code:
def square(n):
    result = n * n
    return result
# Ensure the computed result is returned properly
# Example usage:
print(square(4))  # Output should be 16
print(square(5))  # Output should be 25
print(square(6))  # Output should be 36
```

```
PS C:\Users\Eshwar\OneDrive\Desktop\python> & "C:\Program Files\Python314\python.exe" c:/Users/Eshwar/OneDrive/Desktop/python/ai_coding.py
16
25
36
16
25
36
PS C:\Users\Eshwar\OneDrive\Desktop\python> 
```

Task 3- – IndexError in List Traversal

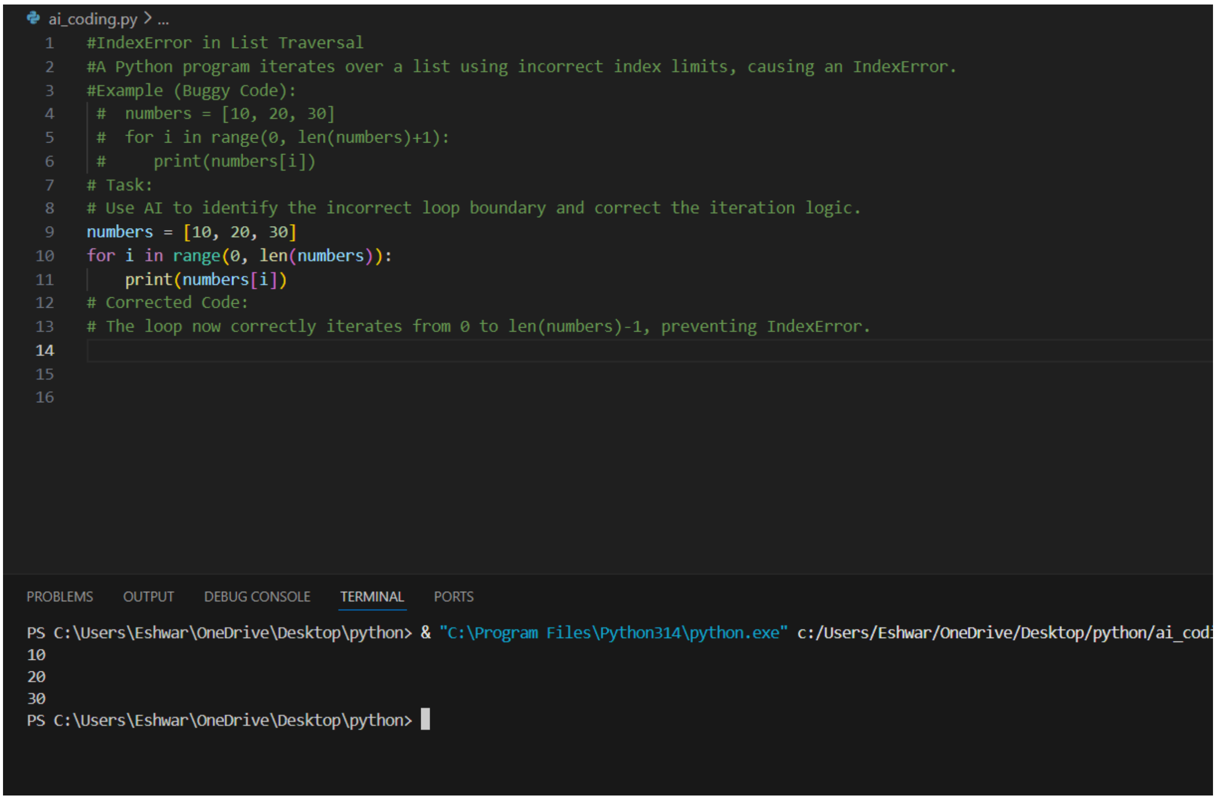
A Python program iterates over a list using incorrect index limits, causing an IndexError.

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.



```python
#IndexError 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.
numbers = [10, 20, 30]
for i in range(0, len(numbers)):
    print(numbers[i])
# Corrected Code:
# The loop now correctly iterates from 0 to len(numbers)-1, preventing IndexError.
```

```
PS C:\Users\Eshwar\OneDrive\Desktop\python> & "C:\Program Files\Python314\python.exe" c:/Users/Eshwar/OneDrive/Desktop/python/ai_cod
10
20
30
PS C:\Users\Eshwar\OneDrive\Desktop\python> 
```

Task 4 – Uninitialized Variable Usage

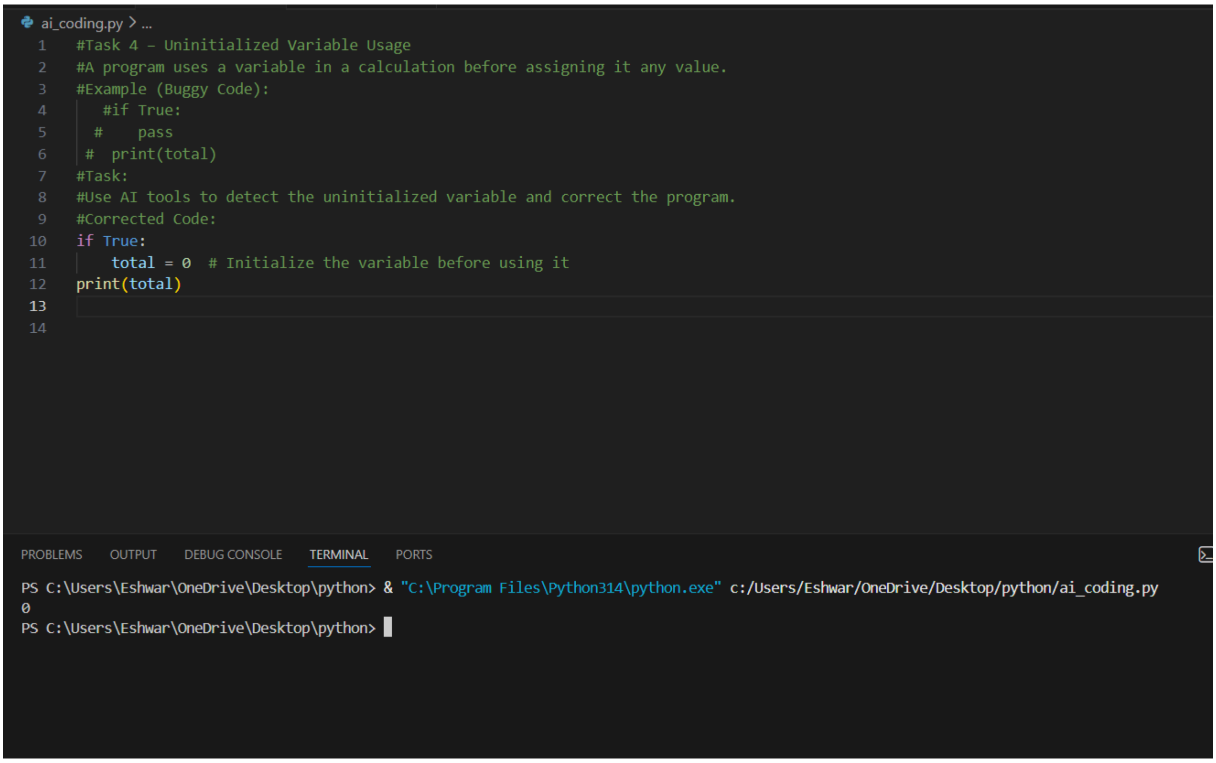
A program uses a variable in a calculation before assigning it any value.

Example (Buggy Code):

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

Task:

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



```python
#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.
#Corrected Code:
if True:
    total = 0  # Initialize the variable before using it
print(total)
```

```
PS C:\Users\Eshwar\OneDrive\Desktop\python> & "C:\Program Files\Python314\python.exe" c:/Users/Eshwar/OneDrive/Desktop/python/ai_coding.py
0
PS C:\Users\Eshwar\OneDrive\Desktop\python> 
```

Task 5 – Logical Error in Student Grading System

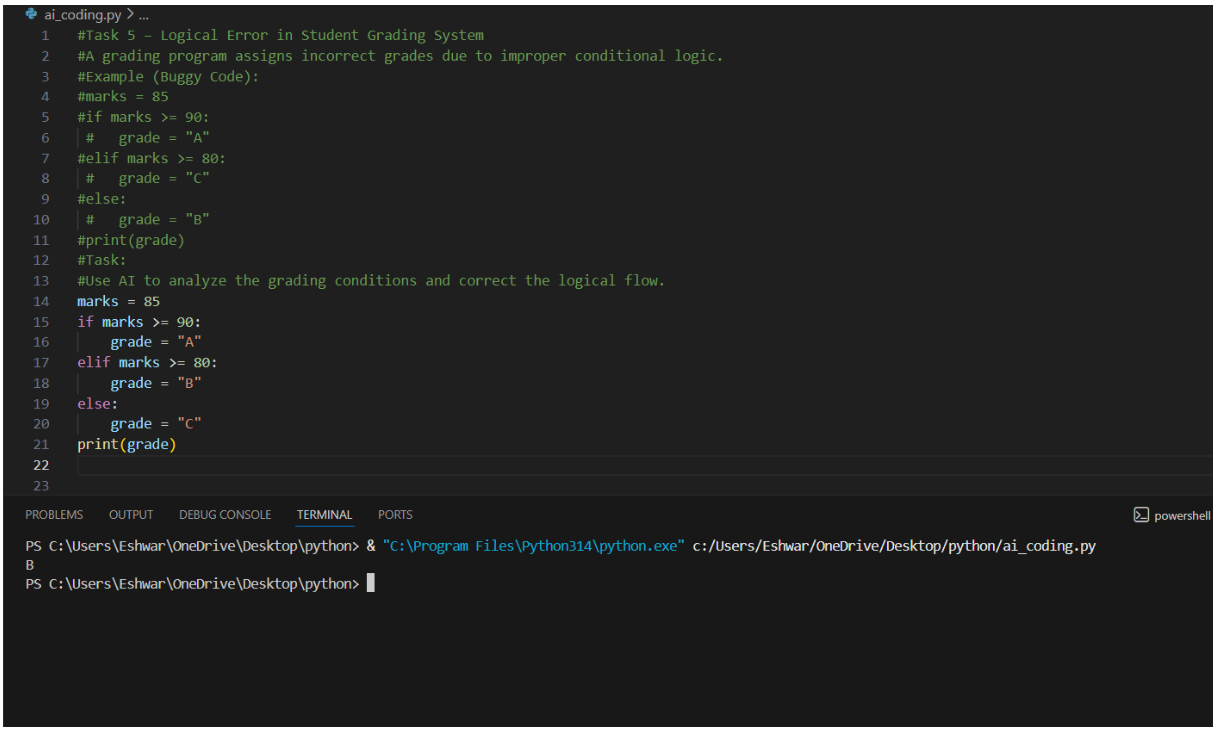
A grading program assigns incorrect grades due to improper conditional logic.

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.



```python
#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.
marks = 85
if marks >= 90:
    grade = "A"
elif marks >= 80:
    grade = "B"
else:
    grade = "C"
print(grade)
```

```
PS C:\Users\Eshwar\OneDrive\Desktop\python> & "C:\Program Files\Python314\python.exe" c:/Users/Eshwar/OneDrive/Desktop/python/ai_coding.py
B
PS C:\Users\Eshwar\OneDrive\Desktop\python> 
```