Assignment – 7.2

RamyaSri

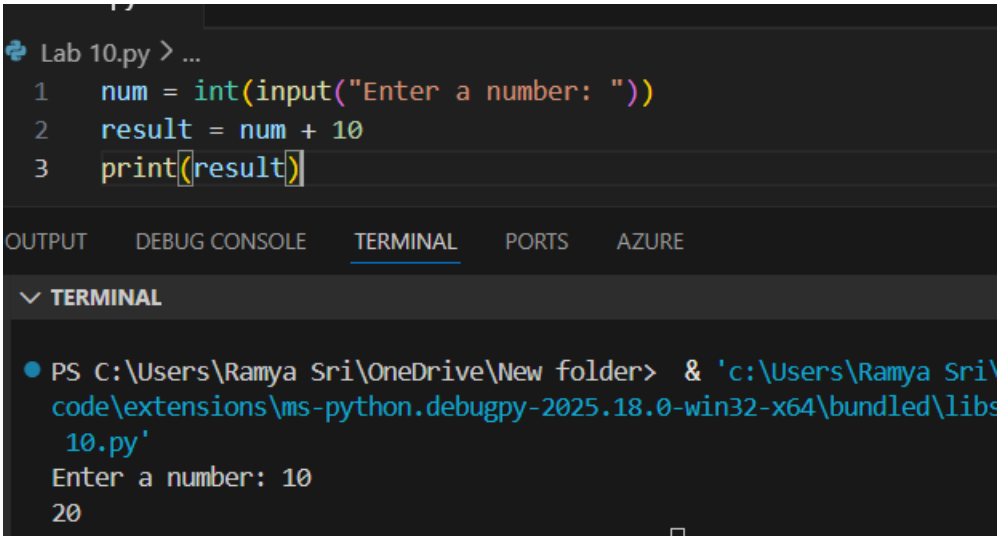
2303A510i9

Task 1 – Runtime Error Due to Invalid Input Type

PROMPT :

Getting a run time error identify the cause for the error and fix it.

CODE :



Justification :

AI helped me understand that the input() function in Python returns a string by default, which caused a TypeError when performing arithmetic operations. By converting the input into an integer using type casting, the runtime error was resolved successfully. This task improved my understanding of data types and their importance in arithmetic operations. It also increased my confidence in identifying and fixing type-related runtime errors.

Task 2 Incorrect Function Return Value

PROMPT :

Not getting an output identify the cause for the error and fix it.

CODE :

```
def square(n):
    result = n * n
    return result
print(square(5))
```

```
OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  AZURE
TERMINAL
PS C:\Users\Ramya Sri\OneDrive\New folder>  & 'c:\Users\Ramya Sri\
code\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs
 10.py'
25
PS C:\Users\Ramya Sri\OneDrive\New folder>
```

Justification :

AI assisted in identifying that the function was missing a return statement, which caused it to return None instead of the expected value. By adding the appropriate return statement, the function was able to correctly provide the computed result. This helped me better understand how functions work internally and why returning values is important. It strengthened my knowledge of function structure and behavior in Python.

Task 3 IndexError in List Traversal

PROMPT :

Getting an index out of range error identify the cause for the error and fix it.

CODE :

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

```
OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  AZURE
TERMINAL
PS C:\Users\Ramya Sri\OneDrive\New folder>  & 'c:\Users\Ramya Sri\
code\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs
 10.py'
10
20
30
PS C:\Users\Ramya Sri\OneDrive\New folder>
```

Justification :

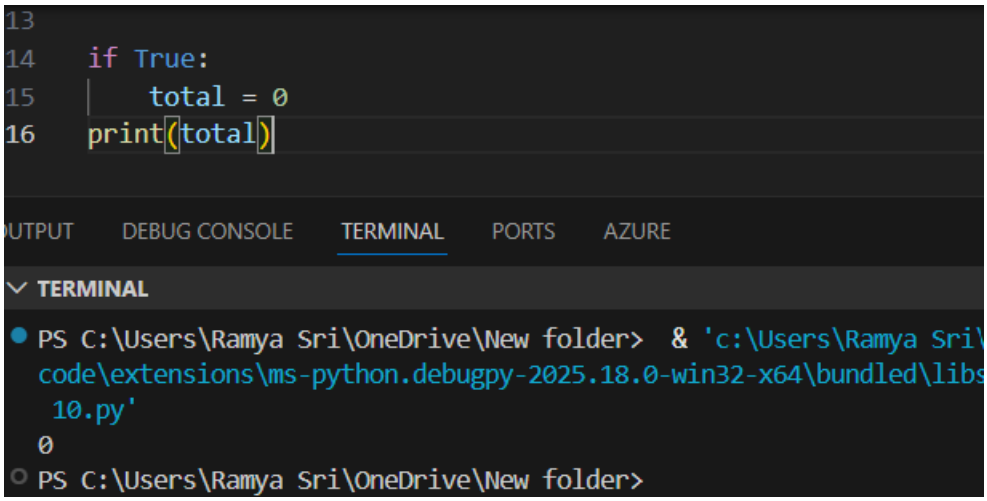
AI detected that the loop boundary condition exceeded the valid index range of the list, which caused an IndexError. By correcting the loop range and removing the extra increment, the program executed without errors. This task helped me understand the importance of proper indexing and boundary conditions when working with lists. It also improved my debugging skills related to iteration logic.

Task 4 Uninitialized Variable Usage

PROMPT :

Getting a name error identify the cause for the error and fix it.

CODE :



Justification :

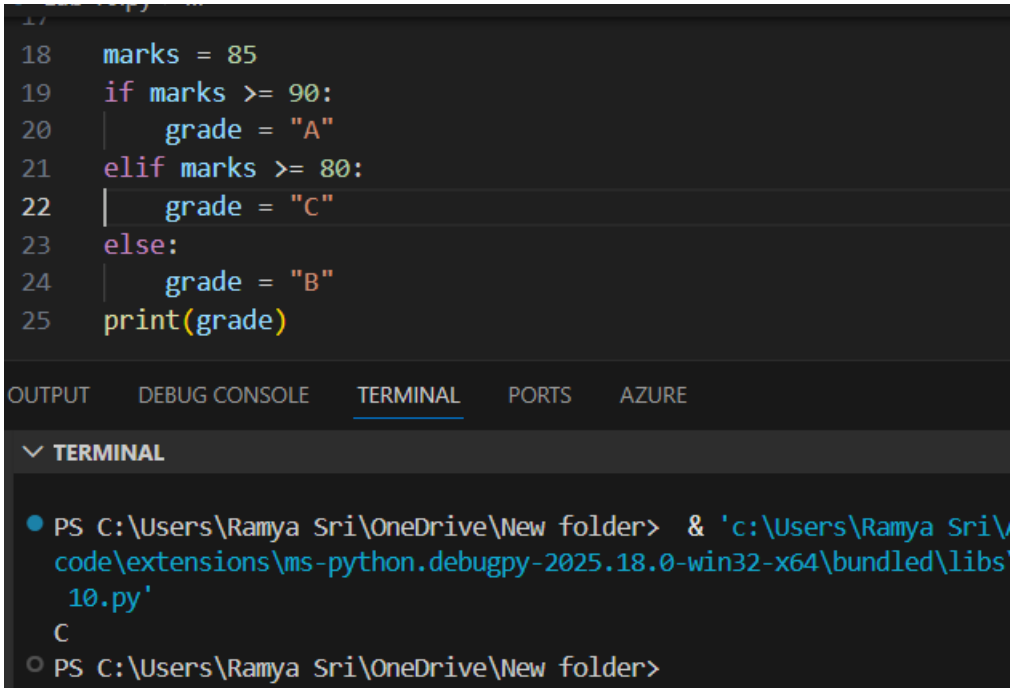
AI identified that a variable was being used before it was initialized, leading to a NameError. By initializing the variable before using it in the program, the error was successfully resolved. This task highlighted the importance of defining variables before usage. It enhanced my understanding of variable scope and initialization in Python.

Task 5 Logical Error in Student Grading System

PROMPT :

Getting wrong output identify the cause for the error and fix it.

CODE :



Justification :

AI analyzed the grading logic and found that the conditional structure was incorrectly assigning grades. By reorganizing and correcting the conditional statements, the program was able to assign accurate grades based on marks. This task improved my understanding of logical flow and condition hierarchy in decision-making statements. It also helped me realize how logical errors can affect program correctness even without syntax issues.