

ASSIGNMENT-7.2

HT.NO:2303A510H6

Batch.No:30

Task 1: Runtime Error Due to Invalid Input Type

Prompt:

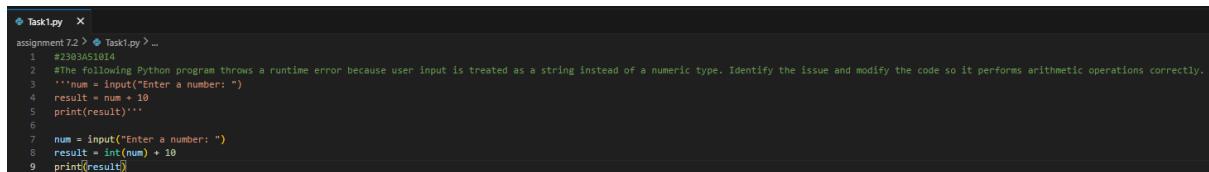
The following Python program throws a runtime error because user input is treated as a string instead of a numeric type. Identify the issue and modify the code so it performs arithmetic operations correctly.

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

result = num + 10

print(result)
```

Code:



A screenshot of a code editor showing a file named Task1.py. The code contains several comments (lines starting with '#') explaining the task. The comments state: '#The following Python program throws a runtime error because user input is treated as a string instead of a numeric type. Identify the issue and modify the code so it performs arithmetic operations correctly.' Below the comments is the original Python code provided in the prompt.

```
Task1.py
assignment 7.2 > Task1.py > ...
1 #The following Python program throws a runtime error because user input is treated as a string instead of a numeric type. Identify the issue and modify the code so it performs arithmetic operations correctly.
2
3 '''num = input("Enter a number: ")'
4 result = num + 10
5 print(result)''
6
7 num = input("Enter a number: ")
8 result = int(num) + 10
9 print(result)
```

Output:



A screenshot of a terminal window showing the execution of Task1.py. The terminal shows the command 'python Task1.py' being run, followed by the user input '3', and the resulting output '13'. The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> C:/miniconda3/Scripts/activate
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> conda activate base
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/Desktop/AI/assignment 7.2/Task1.py"
Enter a number: 3
13
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI>
```

Observation:

The runtime error occurs because `input()` returns a string, and Python cannot add a string to an integer. Converting the input to a numeric type like `int()` fixes the issue and allows the program to execute arithmetic operations correctly.

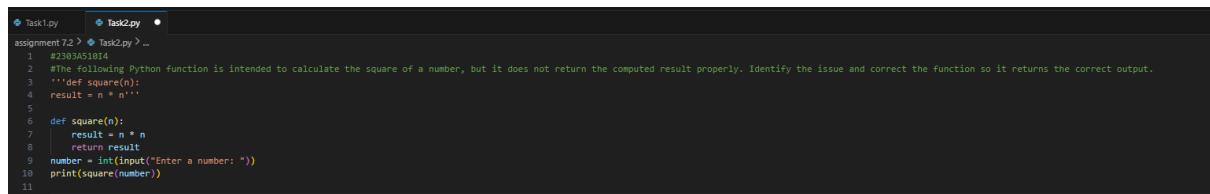
Task 2: Incorrect Function Return Value

Prompt:

The following Python function is intended to calculate the square of a number, but it does not return the computed result properly. Identify the issue and correct the function so it returns the correct output.

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

Code:



```
task2.py Task2.py
assignment 7.2 > Task2.py > ...
1 #Assignment 7.2
2 #The following Python function is intended to calculate the square of a number, but it does not return the computed result properly. Identify the issue and correct the function so it returns the correct output.
3 '''def square(n):
4     result = n * n'''
5
6     def square(n):
7         result = n * n
8         return result
9     number = int(input("Enter a number: "))
10    print(square(number))
11
```

Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + ⌂
● PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> C:/miniconda3/Scripts/activate
● PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> conda activate base
● PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/Desktop/AI/assignment 7.2/Task2.py"
Enter a number: 55
3025
○ PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI>
```

Observation:

The original function calculated the square but did not return the result because the return statement was missing. Adding return result ensures the function outputs the computed value correctly.

Task 3: IndexError in List Traversal

Prompt:

The following Python code causes an IndexError due to incorrect loop boundaries. Identify the issue and correct the iteration logic.

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

```
print(numbers[i])
```

Code:

```
assignment 7.2 > Task3.py > ...
1 #2303A510I4
2 #The following Python code causes an IndexError due to incorrect loop boundaries. Identify the issue and correct the iteration logic.
3 '''numbers = [10, 20, 30]
4 for i in range(0, len(numbers)+1):
5     print(numbers[i])'''
6
7 numbers = [10, 20, 30]
8 for i in range(0, len(numbers)):
9     print(numbers[i])
10 |
```

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> C:/miniconda3/Scripts/activate
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> conda activate base
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/Desktop/AI/assignment 7.2/Task3.py"
● 10
● 20
● 30
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI>
```

Observation:

The error occurred because `range(len(numbers)+1)` exceeds the valid index range. Removing `+1` prevents accessing an out-of-range index and eliminates the `IndexError`.

Task 4: Uninitialized Variable Usage

Prompt:

The following program uses a variable before assigning it a value. Identify the issue and correct the code.

if True:

```
    pass
```

```
    print(total)
```

Code:

```
assignment 7.2 > Task4.py > ...
1  #2303A510I4
2  #The following program uses a variable before assigning it a value. Identify the issue and correct the code.
3  '''if True:
4      pass
5      print(total)'''
6
7  total = 0
8  if True:
9      total = 100
10     print(total)
```

Output:



A screenshot of a terminal window titled "Python". The window shows the following command-line session:

```
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> C:/miniconda3/Scripts/activate
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> conda activate base
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/Desktop/AI/assignment 7.2/Task4.py"
100
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI>
```

Observation:

The variable `total` was used before being initialized, causing a `NameError`. Initializing the variable before usage fixes the issue.

Task 5: Logical Error in Student Grading System

Prompt:

The grading program assigns incorrect grades due to improper conditional logic. Analyze and correct the grading system.

```
marks = 85
```

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

```
    grade = "A"
```

```
elif marks >= 80:
```

```
    grade = "C"
```

```
else:
```

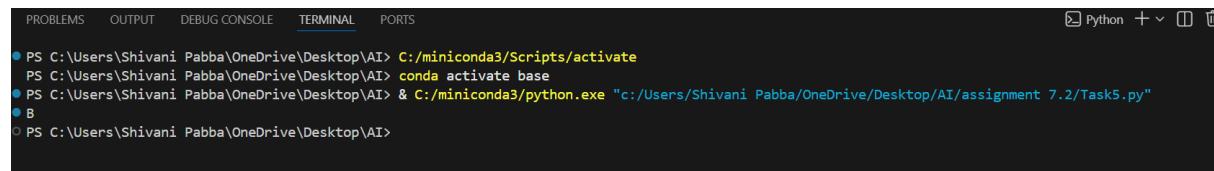
```
    grade = "B"
```

```
print(grade)
```

Code:

```
assignment 7.2 > Task5.py > ...
1  #2303A510I4
2  #The grading program assigns incorrect grades due to improper conditional logic. Analyze and correct the grading system.
3  '''marks = 85
4  if marks >= 90:
5      grade = "A"
6  elif marks >= 80:
7      grade = "C"
8  else:
9      grade = "B"
10 print(grade)'''
11
12 marks = 85
13 if marks >= 90:
14     grade = "A"
15 elif marks >= 80:
16     grade = "B"
17 else:
18     grade = "C"
19 print(grade)
20
```

Output:



A screenshot of a terminal window titled "Python". The window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. The terminal output shows the following command-line session:

```
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> C:/miniconda3/Scripts/activate
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> conda activate base
PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI> & C:/miniconda3/python.exe "c:/Users/Shivani Pabba/OneDrive/Desktop/AI/assignment 7.2/Task5.py"
● B
○ PS C:\Users\Shivani Pabba\OneDrive\Desktop\AI>
```

Observation:

The original logic assigned grade "C" for marks ≥ 80 , which is incorrect. The corrected conditional flow ensures grades are assigned properly based on mark ranges.