

ASSIGNMENT – 7.2

HT NO:2303A51360

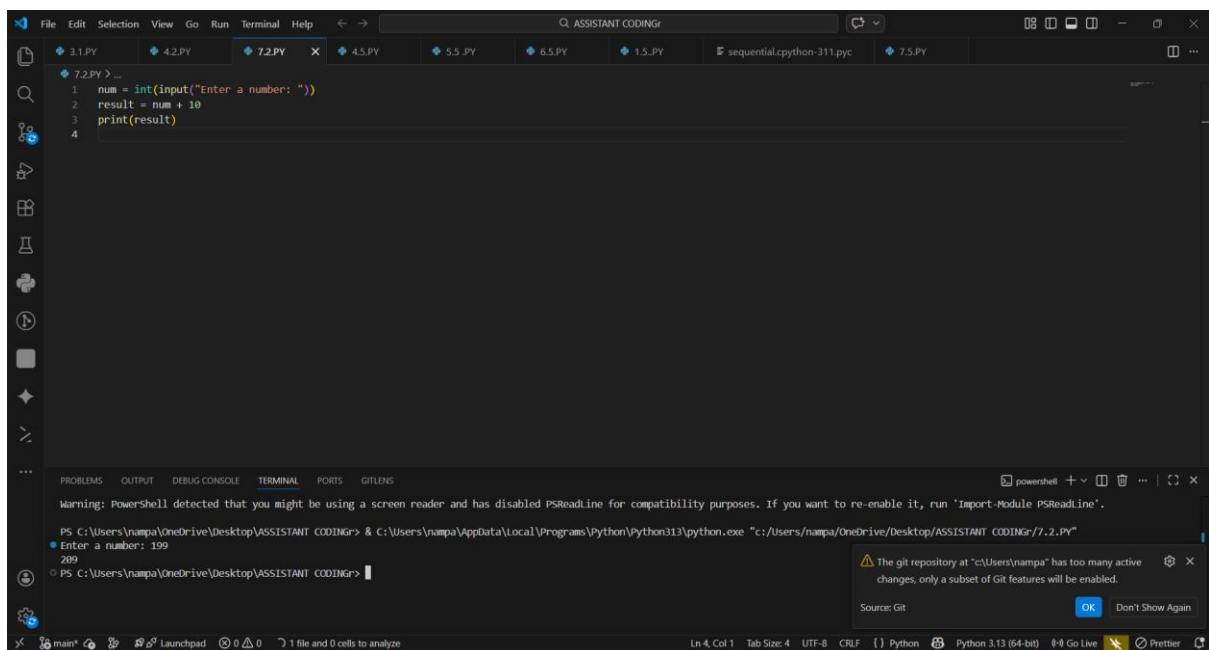
BATCH NO:29

TASK-1:

PROMPT:

```
num = input("Enter a number: ")  
  
result = num + 10  
  
print(result)  
  
#identify the cause of the runtime error and modify  
the program so it executes correctly.
```

CODE:



The screenshot shows the Visual Studio Code editor with a Python file named 7.2.PY. The code in the file is:

```
1 num = int(input("Enter a number: "))  
2 result = num + 10  
3 print(result)  
4
```

The terminal at the bottom shows the command to run the script: `python.exe "c:/Users/nampa/OneDrive/Desktop/ASSISTANT CODING/7.2.PY"`. The output shows the prompt "Enter a number:" followed by the input "199", and then the output "209". A warning message from PowerShell is also visible: "Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadline for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadline'." A notification bubble in the bottom right corner states: "The git repository at 'c:/Users/nampa' has too many active changes, only a subset of Git features will be enabled."

OBSERVATION:

The program takes user input using `input()`, which returns a string by default.

When the code tries to add 10 to this string, a type mismatch occurs, causing a runtime error.

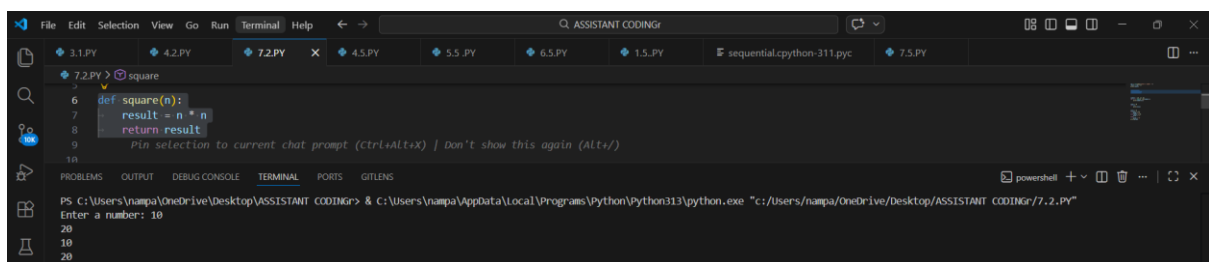
The error happens because arithmetic operations cannot be performed between a string and an integer without converting the input to a numeric type

TASK-2

PROMPT:

```
def square(n):  
    result = n * n  
  
    # Analyze the function and ensure the correct value  
    is returned.  
  
    #fixes the missing return statement and the function  
    returns the correct output.
```

CODE:



OBSERVATION:

In the given function, the value of $n * n$ is calculated and stored in the variable result, but it is never returned from the function.

Because of the missing return statement, the function does not send any value back to the caller and returns None by default.

As a result, the expected output is not produced even though the computation is performed. The bug occurs due to the absence of a return statement in the function definition

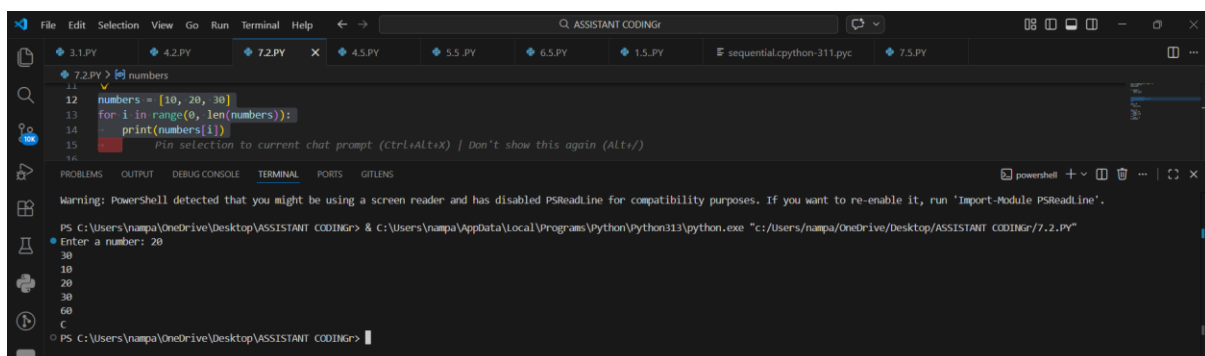
TASK-3

PROMPT:

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

#incorrect loop boundary and correct the iteration logic.

CODE:



The screenshot shows a Visual Studio Code editor window with a Python file named 7.2.PY. The code in the file is:

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

The terminal output shows the execution of the script, which produces the following output:

```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadline for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadline'.
PS C:\Users\nampa\OneDrive\Desktop\ASSISTANT CODING> & C:\Users\nampa\AppData\Local\Programs\Python\Python311\python.exe "c:/Users/nampa/OneDrive/Desktop/ASSISTANT CODING/7.2.PY"
Enter a number: 20
10
20
30
60
C
PS C:\Users\nampa\OneDrive\Desktop\ASSISTANT CODING>
```

OBSERVATION:

The loop runs one step beyond the last valid index because it uses `len(numbers) + 1` as the upper limit. This causes the program to try accessing an index that does not exist in the list, resulting in an `IndexError`. The error occurs due to an incorrect loop boundary that goes out of the list's valid range.

TASK-4

PROMPT:

if True:

pass

`print(total)`

To detect the uninitialized variable and correct the program.

CODE:



The screenshot shows a Visual Studio Code editor window with a Python file named `7.2.PY` open. The code in the file is as follows:

```
17 total = 60
18 if True:
19     print(total)
20
21
```

The terminal window at the bottom shows the command prompt running the script:

```
PS C:\Users\nampa\OneDrive\Desktop\ASSISTANT CODING> & C:\Users\nampa\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/nampa/OneDrive/Desktop/ASSISTANT CODING/7.2.PY"
```

The output of the script is displayed in the terminal:

```
Enter a number: 40
50
10
20
30
60
C
```

OBSERVATION:

In the given program, the variable total is used in the print statement without being assigned any value beforehand.

Since total is never initialized, Python raises a NameError when trying to access it.

The error occurs because the program attempts to use a variable before defining it.

To fix this, the variable must be initialized with a value before it is used in any calculation or output.

TASK-5

PROMPT:

"""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 = "B"
```

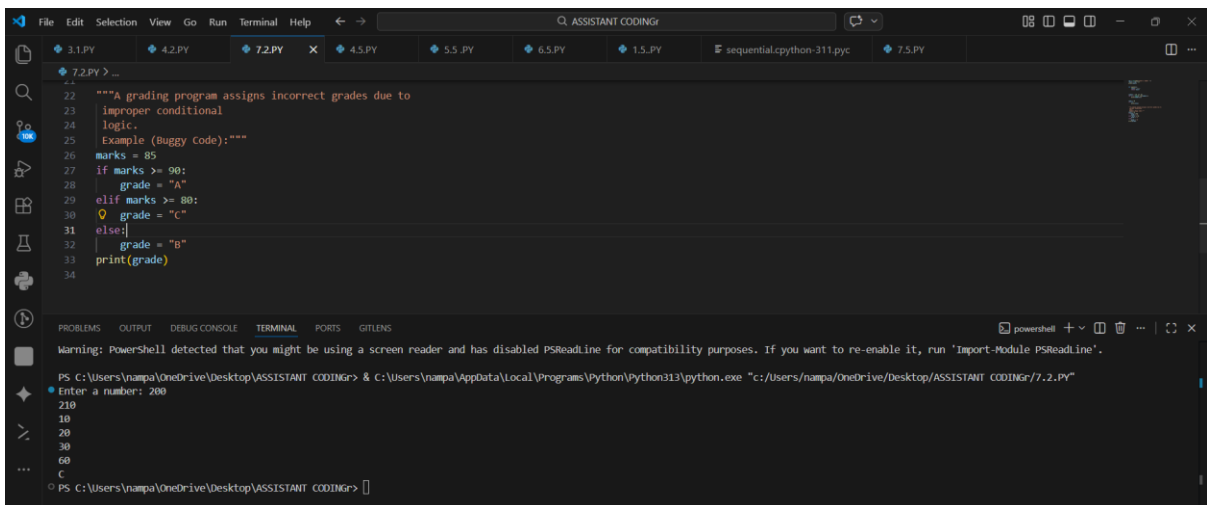
```
else:
```

```
    grade = "c"
```

```
print(grade)
```

#analyze the grading conditions and correct the logical flow.

CODE:



```
22 """A grading program assigns incorrect grades due to
23 improper conditional
24 logic.
25 Example (Buggy Code):"""
26 marks = 85
27 if marks >= 90:
28     grade = "A"
29 elif marks >= 80:
30     grade = "C"
31 else:
32     grade = "B"
33 print(grade)
34
```

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\nampa\OneDrive\Desktop\ASSISTANT CODING> & C:\Users\nampa\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/nampa/OneDrive/Desktop/ASSISTANT CODING/7.2.PY"

Enter a number: 200

210

10

20

30

60

...

C

OBSERVATION:

In the given program, the grading conditions are logically incorrect.

For marks greater than or equal to 80, the grade is assigned as "C"

instead of a higher grade, and the else block assigns "B" for lower marks.

This causes wrong grade assignment because higher marks

should correspond to better grades. The logical flow of conditions is reversed,

leading to incorrect output. The issue occurs due to improper ordering and incorrect grade mapping in the conditional statements.