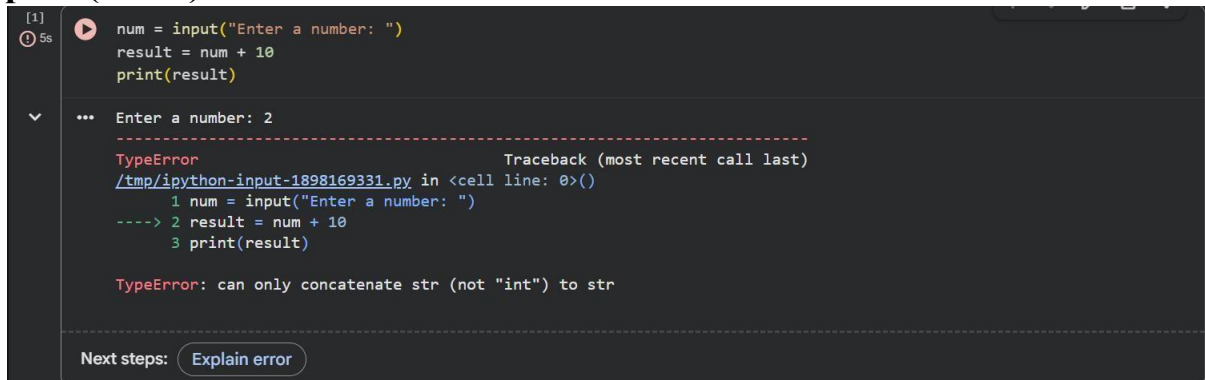

Lab Assignment # 7.2

Program : B. Tech (CSE)
Specialization :
Course Title : AI Assisted Coding
Course Code : 23CS002PC304
Semester II
Academic Session : 2025-2026
Name of Student : P.Harshavardhan
Enrollment No. : 2403A51L37
Batch No. : 52
Date : 30-01-2026

Task 1 – Runtime Error Due to Invalid Input Type

(Buggy Code): num =
input("Enter a number: ") result
= num + 10
print(result)



```
[1] 5s num = input("Enter a number: ")
result = num + 10
print(result)

... Enter a number: 2

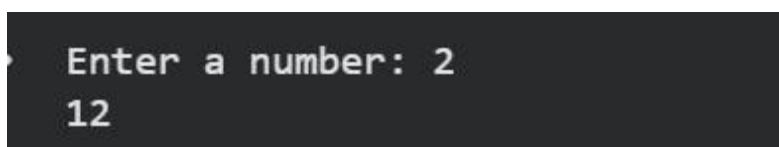
-----
TypeError                                Traceback (most recent call last)
/tmp/ipython-input-1898169331.py in <cell line: 0>()
      1 num = input("Enter a number: ")
----> 2 result = num + 10
      3 print(result)

TypeError: can only concatenate str (not "int") to str

Next steps: Explain error
```

Output:

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

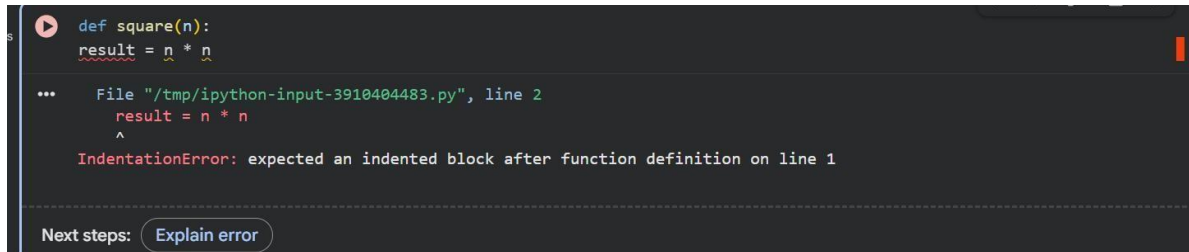


```
Enter a number: 2
12
```

Task 2 – Incorrect Function Return Value

(Buggy Code):

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

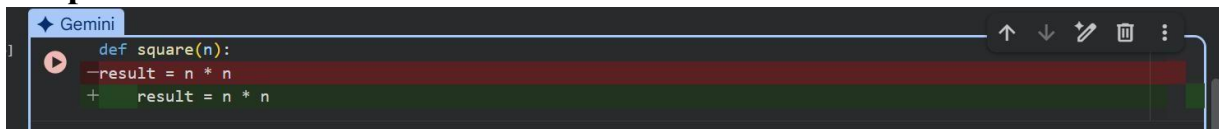


```
def square(n):
result = n * n

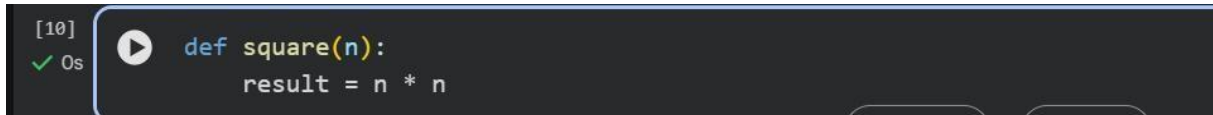
... File "/tmp/ipython-input-3910404483.py", line 2
      result = n * n
      ^
IndentationError: expected an indented block after function definition on line 1
```

Next steps: [Explain error](#)

Output:



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

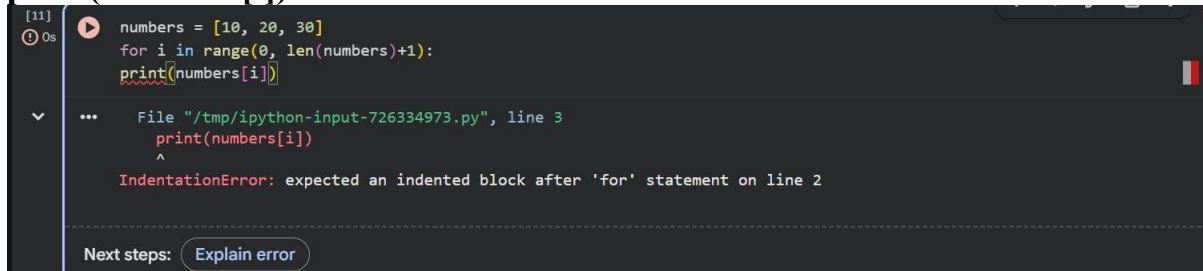


```
[10] def square(n):
      result = n * n
```

✓ 0s

Task-3 Index Error in List Traversal

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

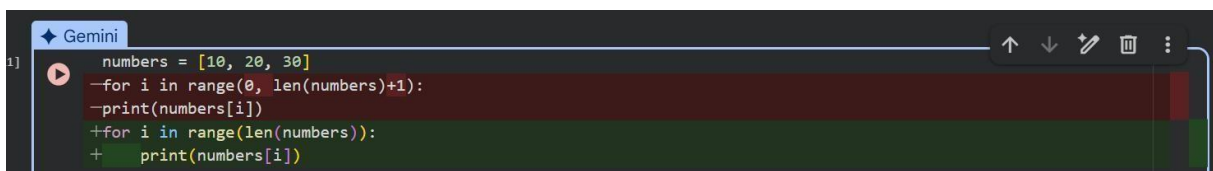


```
numbers = [10, 20, 30]
for i in range(0, len(numbers)+1):
print(numbers[i])

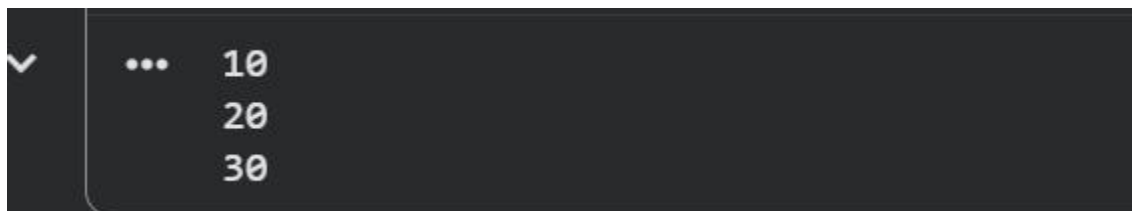
... File "/tmp/ipython-input-726334973.py", line 3
      print(numbers[i])
      ^
IndexError: list index out of range
```

Next steps: [Explain error](#)

Output:



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



```
... 10
      20
      30
```

Task 4 – Uninitialized Variable Usage

(Buggy Code):

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

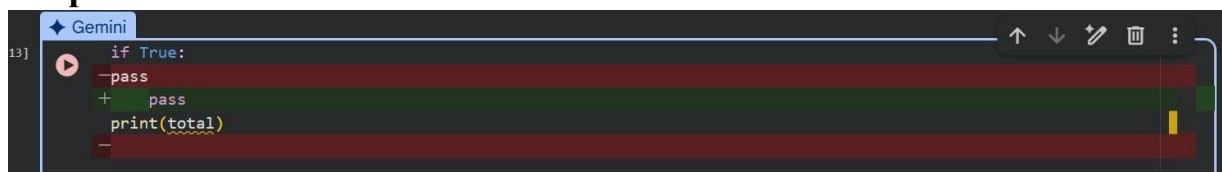


```
[13] 0s if True:
      pass
      print(total)

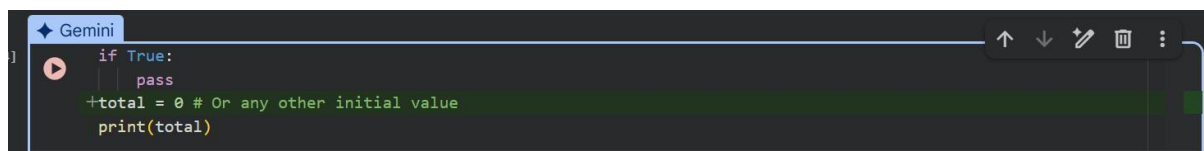
... File "/tmp/ipython-input-1170978020.py", line 2
      pass
      ^
IndentationError: expected an indented block after 'if' statement on line 1

Next steps: Explain error
```

Output:



```
[13] Gemini if True:
           pass
           print(total)
```



```
Gemini if True:
        pass
+total = 0 # Or any other initial value
print(total)
```



```
... 0
```

Task 5 – Logical Error in Student Grading System

(Buggy Code): marks = 85

if marks >= 90: grade =

"A" elif marks >= 80:

grade = "C" else:

grade = "B"

print(grade)



```
[16] 0s marks = 85
      if marks >= 90:
          grade = "A"
      elif marks >= 80:
          grade = "C"
      else:
          grade = "B"
      print(grade)

... File "/tmp/ipython-input-2691675298.py", line 3
          grade = "A"
          ^
IndentationError: expected an indented block after 'if' statement on line 2

Next steps: Explain error
```

Output:



A screenshot of a code editor window titled "Gemini". The code is written in Python and defines a variable "marks" with the value 85. It then uses an if-elif-else structure to assign a grade to the variable "grade". The conditions are: if marks are greater than or equal to 90, grade is "A"; if marks are greater than or equal to 80, grade is "C"; otherwise, grade is "B". Finally, the grade is printed. The code is syntax-highlighted with red for minus signs, green for plus signs, and yellow for the print statement. The editor has a dark theme and a toolbar with icons for undo, redo, and search.

```
marks = 85
if marks >= 90:
    -grade = "A"
    + grade = "A"
elif marks >= 80:
    -grade = "C"
    + grade = "C"
else:
    -grade = "B"
    + grade = "B"
print(grade)
```

... C