

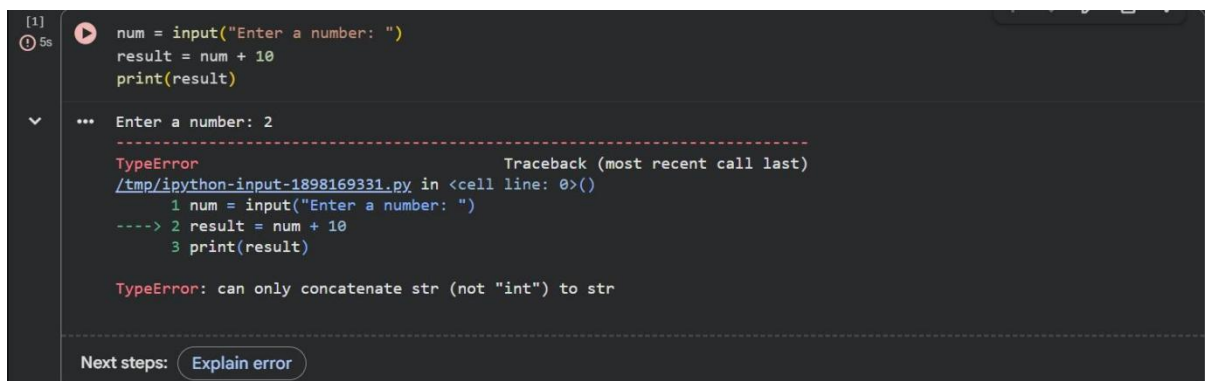
School of Computer Science and Artificial Intelligence

Lab Assignment # 7.2

Program	: B. Tech (CSE)
Specialization	:
Course Title	: AI Assisted Coding
Course Code	: 23CS002PC304
Semester	II
Academic Session	: 2025-2026
Name of Student	: Rohan Beri
Enrollment No.	: 2403A51L09
Batch No.	: 51
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] num = input("Enter a number: ")
    result = num + 10
    print(result)

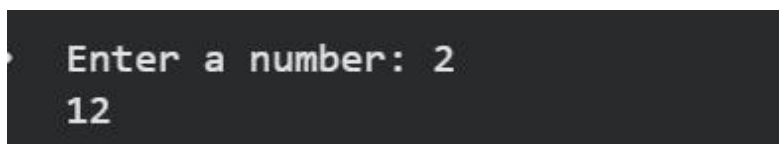
*** Enter a number: 2
-----
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-1898169331.py in <cell line: 0>()
      1 num = input("Enter a number: ")
----> 2 result = num + 10
      3 print(result)

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)
```

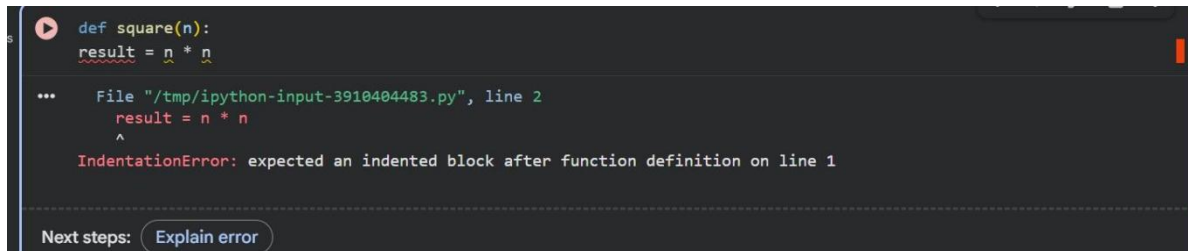


```
Enter a number: 2
12
```

Task 2 – Incorrect Function Return Value

(Buggy Code):

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

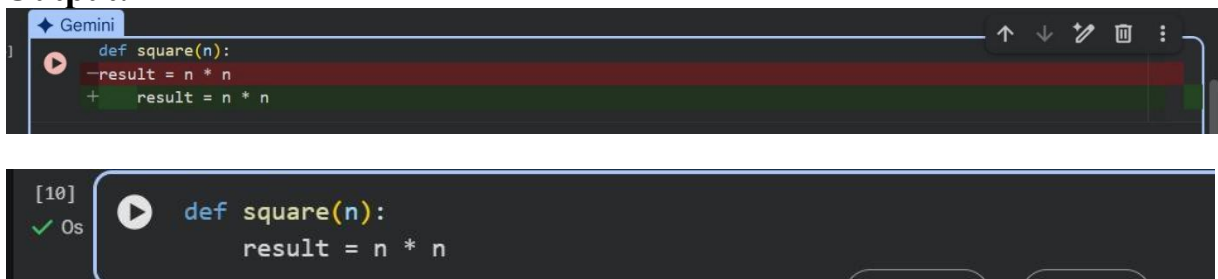


```
def square(n):
    result = n * n

... 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
```

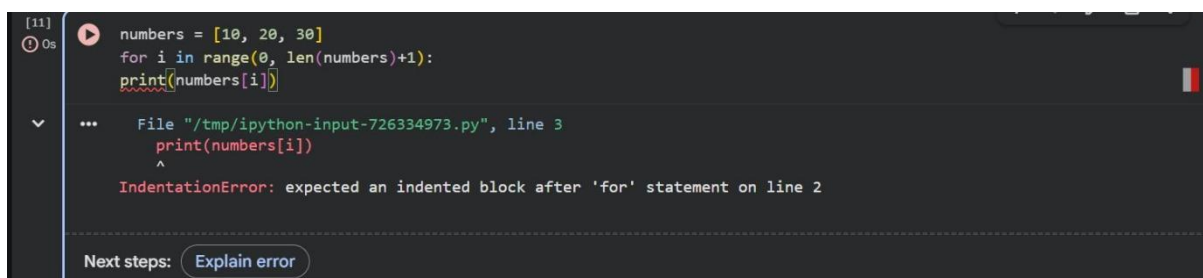
[10] ✓ 0s

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

Task-3 Index Error in List Traversal

(Buggy Code):

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

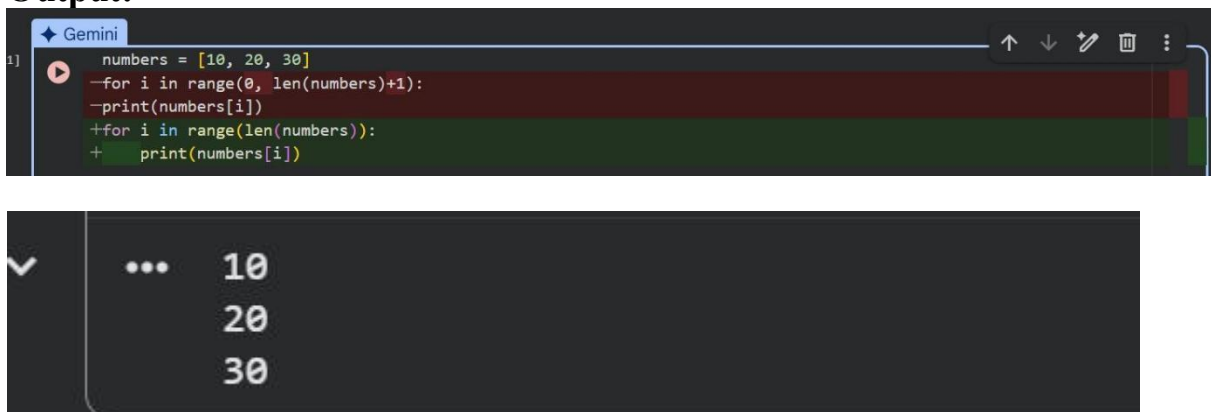


```
numbers = [10, 20, 30]
for i in range(0, len(numbers)+1):
    print(numbers[i])

... 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
```

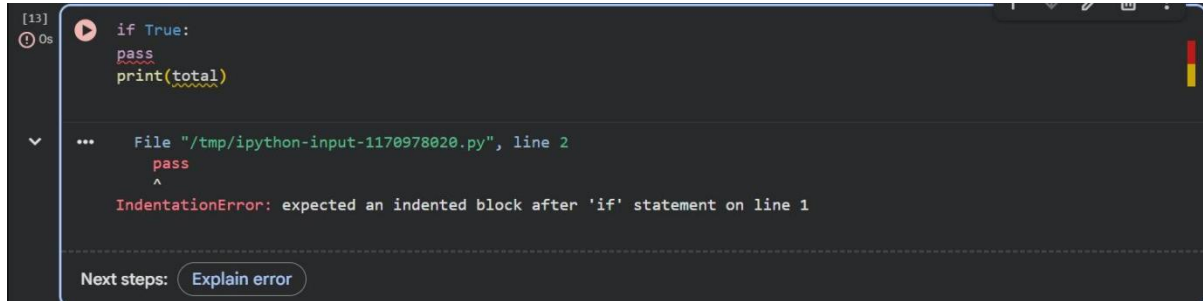
Task 4 – Uninitialized Variable Usage

(Buggy Code):

if True:

pass

print(total)



The screenshot shows a Jupyter Notebook cell with the following code:

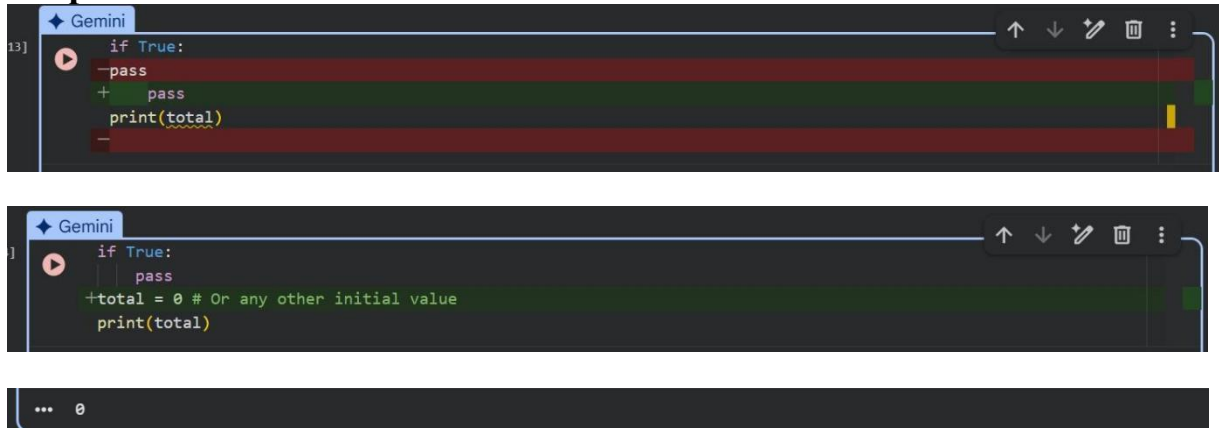
```
[13] In [13]: if True:
            pass
            print(total)
```

The code is executed, and an error is displayed:

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

Below the error message, there is a "Next steps:" section with a button labeled "Explain error".

Output:



The first screenshot shows the original code with red squiggly lines indicating errors:

```
[13] In [13]: if True:
            -pass
            + pass
            print(total)
```

The second screenshot shows the corrected code with a new line added to initialize the variable:

```
[13] In [13]: if True:
            pass
            +total = 0 # Or any other initial value
            print(total)
```

Below the code editor, there is a "..." button and a "0" indicating no output.

Task 5 – Logical Error in Student Grading System

(Buggy Code):

marks = 85

if marks >= 90: grade = "A"

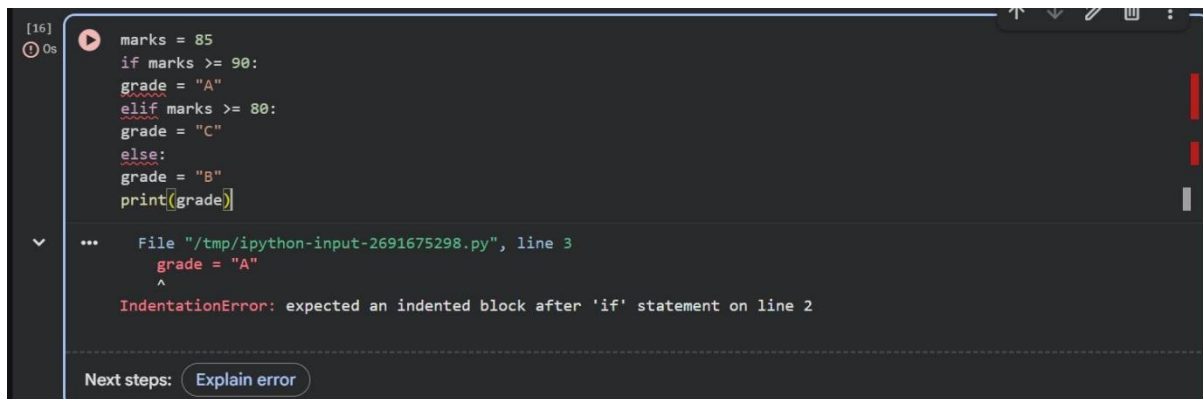
elif marks >=

80: grade = "C"

else:

grade="B"

print(grade)



The screenshot shows a code editor with the following Python code:

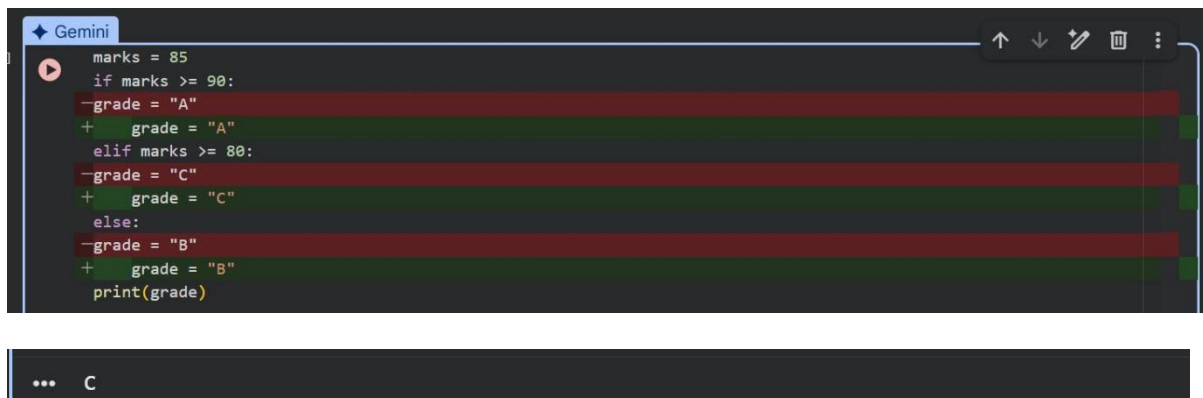
```
marks = 85
if marks >= 90:
    grade = "A"
elif marks >= 80:
    grade = "C"
else:
    grade = "B"
print(grade)
```

Below the code, an error message is displayed:

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

At the bottom, there is a button labeled "Explain error".

Output:



The screenshot shows the same Python code as above, but with a logical error highlighted in red. The code is:

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

Below the code, the output is displayed as:

```
... C
```