School of Computer Science and Artificial Intelligence

LabAssignment#7.2

Program :B.Tech(CSE)

Specialization :AIML

Course Title :AI Assisted Coding

Course Code : 23CS002PC304

Semester : VI

AcademicSession :2025-2026

Name of Student :P.Sushanthi

Enrollment No. : 2303A52102

Batch No. : 33

Date :3/2/2026

Task–1: Runtime Error Due to Invalid Input Type

Buggy Code:

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

```
Enter a number: 9
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-671842994.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
```

AI Explanation:

The program attempts to perform arithmetic on a string value. The input must be converted to an integer or float before calculations.

Corrected Code:

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

```
Enter a number: 9
19
```

Conclusion:

AI identified the runtime error and resolved it by converting user input into a numeric type.

Task 2 – Incorrect Function Return Value:

Buggy code:

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

```
Enter a number: 9
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-671842994.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
```

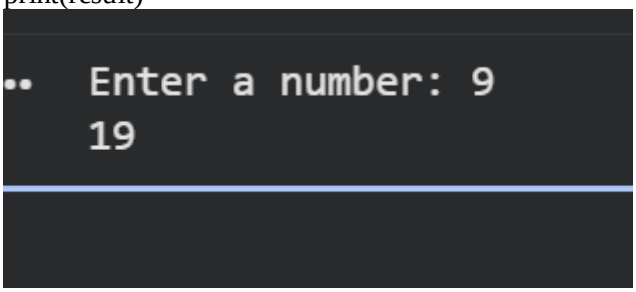
AI Explanation:

The program attempts to perform arithmetic on a string value. The input must be

converted to an integer or float before calculations.

Corrected Code:

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



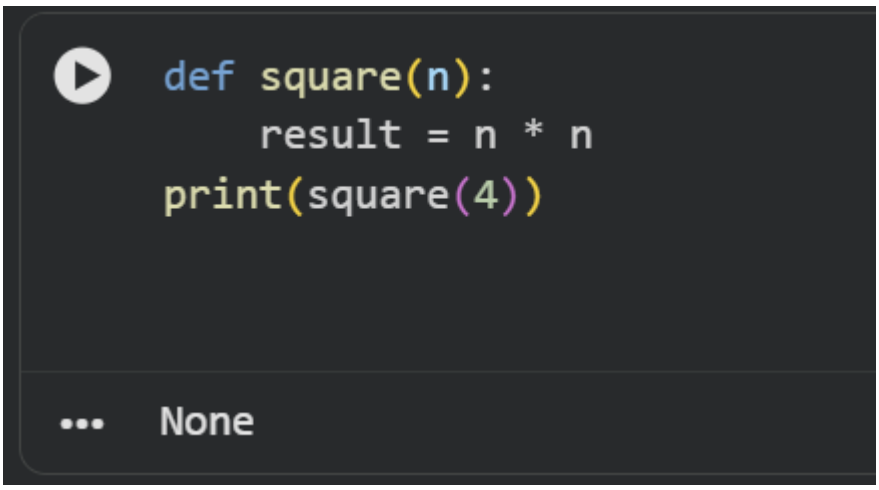
Conclusion:

AI identified the runtime error and resolved it by converting user input into a numeric type.

Task 2 – Incorrect Function Return Value:

Buggy Code:

```
def square(n):
    result = n * n
print(square(4))
```



AI Explanation:

Without a return statement, the computed value cannot be accessed outside the function.

Corrected Code:

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

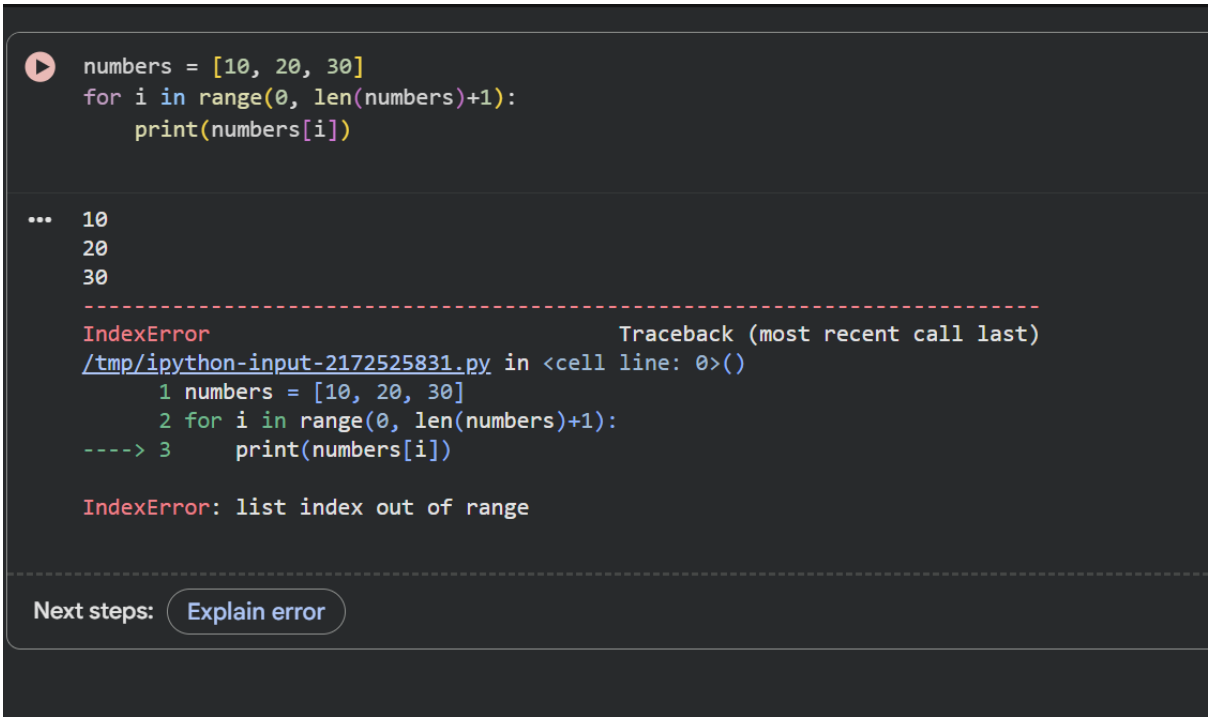
```
16
```

Conclusion:

AI detected the missing return statement and corrected the function behavior.

Task 3 – IndexError in List Traversal

Buggy Code:



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

```
10
20
30
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
/tmp/ipython-input-2172525831.py in <cell line: 0>()
      1 numbers = [10, 20, 30]
      2 for i in range(0, len(numbers)+1):
----> 3     print(numbers[i])

IndexError: list index out of range
```

Next steps: Explain error

AI Explaination:

Python list indices range from 0 to len(list) - 1. The loop condition must be corrected.

Corrected Code:

```
[11]
✓ 0s
numbers = [10, 20, 30]
for i in range(0, len(numbers)):
    print(numbers[i])

10
20
30
```

Conclusion:
AI corrected the loop boundary to prevent out-of-range access.

Task 4 – Uninitialized Variable Usage

Buggy Code:

```
if True:
    pass
print(total)

0
```

AI Explanation:
Variables must be initialized before usage, even if logic blocks are present.

Corrected Code:

```
total = 5
if True:
    pass
print(total)

5
```

Conclusion:
AI detected the uninitialized variable and ensured safe initialization.

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)
```

```
C
```

AI Explanation:
Grading conditions must follow descending or correct logical ranges.

Corrected Code:

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

```
B
```

Conclusion:
AI corrected the logical flow to ensure accurate grade assignment.