

## 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 : Kothapally Srikanth  
Enrollment No. : 2403A51L42  
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)
```

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

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

      ... Enter a number: 2
```

A tooltip indicates a **TypeError** at the line `result = num + 10`. The error message is: `TypeError: can only concatenate str (not "int") to str`. The cell has a status bar showing "0s".

**Output:**

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

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

The screenshot shows the output of the corrected code:

```
• Enter a number: 2
12
```

## Task 2 – Incorrect Function Return Value

(Buggy Code):

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

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

```
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
```

Below the code, there is a button labeled "Next steps: Explain error".

Output:

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

```
[10] 0s  def square(n):
        result = n * n
```

Below the code, there is a button labeled "Next steps: Explain error".

## Task-3 Index Error in List Traversal

(Buggy Code):

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

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

```
[11] 0s  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])
            ^
IndentationError: expected an indented block after 'for' statement on line 2
```

Below the code, there is a button labeled "Next steps: Explain error".

Output:

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

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

Below the code, there is a button labeled "Next steps: Explain error".

The output window shows the following output:

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

## Task 4 – Uninitialized Variable Usage

(Buggy Code):

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

A screenshot of an IDE interface showing a Python script. The code is:

```
[13] ① 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
```

The line with the error is highlighted in red. A tooltip "Next steps: Explain error" is visible at the bottom.

Output:

Two screenshots of an IDE interface showing the same Python script after modification. The first screenshot shows the initial state with the error:

```
[13] ① Gemini  
if True:  
    -pass  
    + pass  
    print(total)  
-
```

The second screenshot shows the code after adding an initialization line:

```
[13] ① Gemini  
if True:  
    || pass  
    +total = 0 # Or any other initial value  
    | print(total)
```

The corrected code is now highlighted in green.

## 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 interface with a dark theme. On the left, there's a sidebar with a play button icon and the number '16'. The main area contains 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 of the editor, there is a button labeled "Next steps: Explain error".

Output:

The screenshot shows a code editor interface with a dark theme, similar to the one above. The top part displays the same Python code as before, but with syntax highlighting and some changes in the code itself. The code now looks like this:

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

The bottom part of the interface shows the output of the code execution:

```
...   C
```