

## 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 : R.Sai Kowshik  
Enrollment No. : 2403A51L23  
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)
```

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

```
[1] ① 5s ▶ num = input("Enter a number: ")
      result = num + 10
      print(result)
  ↴ ... Enter a number: 2
  -----
  TypeError: can only concatenate str (not "int") to str
  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:**

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

```
▶ -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
```

The screenshot shows an IDE interface with a dark theme. A code editor window displays the following Python 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 editor, a button labeled "Next steps: Explain error" is visible.

Output:

The screenshot shows an IDE interface with a dark theme. A code editor window displays the following Python code:

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

Below the code editor, a terminal window shows the output of the code execution:

```
[10] 0s  ... 10
20
30
```

## 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 an IDE interface with a dark theme. A code editor window displays the following Python 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 editor, a button labeled "Next steps: Explain error" is visible.

Output:

The screenshot shows an IDE interface with a dark theme. A code editor window displays the following Python code:

```
[1]  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 editor, a terminal window shows the output of the code execution:

```
[1]  ... 10
20
30
```

## Task 4 – Uninitialized Variable Usage

(Buggy Code):

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

A screenshot of an IDE interface showing a code editor and a terminal. The code in the editor 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 terminal below shows the error message:

```
IndentationError: expected an indented block after 'if' statement on line 1
```

Next steps: Explain error

Output:

Two screenshots of an IDE interface showing the execution of the code. The first screenshot shows the initial state:

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

The second screenshot shows the result of the execution:

```
[13] ① 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)
```

The screenshot shows a code editor window with a Python script. The code defines a variable `marks` and sets it to 85. It then uses an `if` statement to check if `marks` is greater than or equal to 90, in which case it sets `grade` to "A". Below this, there is an `elif` block that is not properly indented, resulting in a syntax error. The code editor highlights the `elif` keyword in red and shows an error message below the code: `IndentationError: expected an indented block after 'if' statement on line 2`. At the bottom of the window, there is a button labeled `Explain error`.

Output:

The screenshot shows a code editor window titled "Gemini" with the same Python script as the previous screenshot. The code is identical, including the syntax error. The output of the script is displayed below the code editor, showing the result: `*** C`. The word `C` is highlighted in green, indicating it is the output of the `print(grade)` statement.