

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	: I.Sathwik Rajeshwara Chary
Enrollment No.	: 2403A51L03
Batch No.	: 51
Date	: 30/01/26

Submission Starts here**Screenshots:****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] In [55]: num = input("Enter a number: ")
result = num + 10
print(result)
```

When the code is run, it prompts the user for input:

```
...> Enter a number: 2
```

However, the input is a string ('2') instead of an integer. This causes a `TypeError`:

```
-----  
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:

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

```
[1] In [55]: num = input("Enter a number: ")
+num = int(input("Enter a number: "))
result = num + 10
print(result)
```

When the code is run, it prompts the user for input:

```
> Enter a number: 2
```

The output is:

```
12
```

Task 2 – Incorrect Function Return Value

(Buggy Code):

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

The screenshot shows a code editor with a dark theme. A tooltip is displayed over the line 'result = n * n'. The tooltip contains the following text:
```py  
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 tooltip, there is a button labeled 'Next steps: Explain error'.

Output:

The screenshot shows a code editor with a dark theme. The code has been corrected:
```py  
def square(n):  
 result = n \* n  
+ result = n \* n  
```  
A tooltip is shown above the first 'result' line, indicating a conflict with the previous assignment. Below the code, the output window shows the results of the function calls.

Task 3 – IndexError 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 with a dark theme. A tooltip is displayed over the line 'print(numbers[i])'. The tooltip contains the following text:
```py  
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 tooltip, there is a button labeled 'Next steps: Explain error'.

Output:

The screenshot shows a code editor with a dark theme. The code has been corrected:
```py  
numbers = [10, 20, 30]  
for i in range(0, len(numbers)+1):  
 print(numbers[i])  
+for i in range(len(numbers)):  
+ print(numbers[i])  
```  
A tooltip is shown above the first 'print' line, indicating a conflict with the previous assignment. Below the code, the output window shows the results of the function calls.

Output window:
```text  
... 10  
20  
30  
```

Task 4 – Uninitialized Variable Usage

(Buggy Code):

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

A screenshot of a code editor window titled '[13]'. The code is:

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

The line 'print(total)' is underlined with a red squiggle. The status bar at the bottom says 'Next steps: Explain error'.

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

Output:

Two screenshots of a code editor window titled 'Gemini'.

The first screenshot shows the corrected code:

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

The second screenshot shows the output of the code:

```
] 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 interface with a dark theme. In the top left corner, there is a status bar with '[16] Os'. The main area contains a Python script:

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

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

Output:

The screenshot shows a code editor interface with a dark theme. In the top left corner, there is a status bar with 'Gemini'. The main area contains the same Python script as the previous screenshot, but it has been corrected:

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

Below the script, the output 'C' is displayed.