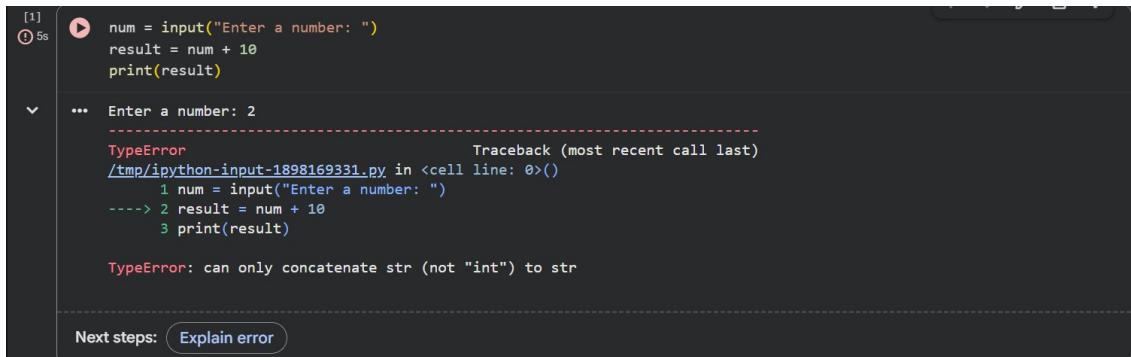


School of Computer Science and Artificial Intelligence**Lab Assignment # 7.2**

| | |
|-------------------------|----------------------|
| Program | : B. Tech (CSE) |
| Specialization | : - |
| Course Title | : AI Assisted Coding |
| Course Code | : 23CS002PC304 |
| Semester | : II |
| Academic Session | : 2025-2026 |
| Name of Student | : Vemula Rikith |
| Enrollment No. | : 2403A51L35 |
| Batch No. | : 52 |
| 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)
```



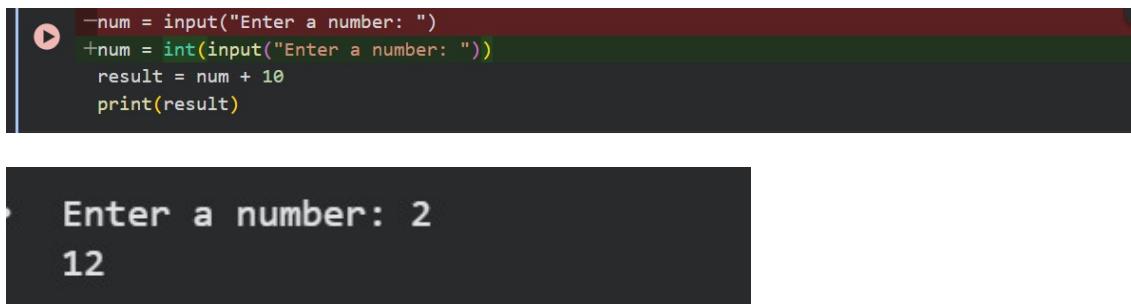
The screenshot shows a Jupyter Notebook cell with the following code:

```
[1] ① 5s ▶ num = input("Enter a number: ")
      result = num + 10
      print(result)

...  Enter a number: 2
-----  
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
```

When the user enters '2' as input, a `TypeError` is raised because it cannot concatenate a string ('2') with an integer (10). A red box highlights the error message: `TypeError: can only concatenate str (not "int") to str`. Below the code cell, there is a button labeled `Explain error`.

Output:

The screenshot shows the same Jupyter Notebook cell after the code has been corrected:

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

The user's input '2' is shown in a red box at the bottom of the cell. The output shows the corrected code and the resulting value '12'.

Task 2 – Incorrect Function Return Value

(Buggy Code):

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

A screenshot of an IDE interface. The code editor shows a Python file with the following content:

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

```

The line "result = n * n" is highlighted in red, indicating a syntax error. A tooltip or status bar at the bottom says "IndentationError: expected an indented block after function definition on line 1".

Output:

The code editor shows the same Python file, but the error has been fixed. The line "result = n * n" now has the correct indentation:

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

The IDE shows a green checkmark icon and the text "0s" next to the play button, indicating the code is now valid.

Task 3 – IndexError in List Traversal

(Buggy Code):

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

A screenshot of an IDE interface. The code editor shows a Python file with the following content:

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

The line "print(numbers[i])" is highlighted in red, indicating an error. A tooltip or status bar at the bottom says "IndentationError: expected an indented block after 'for' statement on line 2".

Output:

The code editor shows the same Python file, but the error has been fixed. The line "print(numbers[i])" now has the correct indentation:

```
for i in range(0, len(numbers)):
    print(numbers[i])
```

The IDE shows a green checkmark icon and the text "0s" next to the play button, indicating the code is now valid. Below the code editor, the terminal window shows the output of the corrected code:

```
10
20
30
```

Task 4 – Uninitialized Variable Usage

(Buggy Code):

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

The screenshot shows an IDE interface with a code editor and a terminal or output window. The code in the editor is:

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

The terminal window below shows the error:

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

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

Output:

The screenshot shows the IDE after the code has been corrected. The code in the editor is:

```
[13] ② if True:  
    -pass  
    + pass  
    print(total)  
    -
```

A specific line, "print(total)", is highlighted in green.

The screenshot shows the IDE with the final corrected code. The code in the editor is:

```
[13] ③ if True:  
    pass  
    +total = 0 # Or any other initial value  
    print(total)
```

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)
```

```
[16] ⓘ 0s
  ◀ marks = 85
    if marks >= 90:
      grade = "A"
    elif marks >= 80:
      grade = "C"
    else:
      grade = "B"
    print(grade)

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

Next steps: Explain error
```

Output:

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

...
  ... C
```