

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
ProgramName: B. Tech		Assignment Type: Lab	AcademicYear:2025-2026
CourseCoordinatorName		Venkataramana Veeramsetty	
Instructor(s)Name		Dr. V. Venkataramana (Co-ordinator)	
		Dr. T. Sampath Kumar	
		Dr. Pramoda Patro	
		Dr. Brij Kishor Tiwari	
		Dr.J.Ravichander	
		Dr. Mohammand Ali Shaik	
		Dr. Anirodh Kumar	
		Mr. S.Naresh Kumar	
		Dr. RAJESH VELPULA	
		Mr. Kundhan Kumar	
		Ms. Ch.Rajitha	
		Mr. M Prakash	
		Mr. B.Raju	
		Intern 1 (Dharma teja)	
		Intern 2 (Sai Prasad)	
		Intern 3 (Sowmya)	
		NS_2 (Mounika)	
CourseCode	24CS002PC215	CourseTitle	AI Assisted Coding
Year/Sem	II/I	Regulation	R24
Date and Day of Assignment	Week4 - Wednesday	Time(s)	
Duration	2 Hours	Applicable to Batches	
AssignmentNumber: 7.3(Present assignment number)/24(Total number of assignments)			
Q.No.	Question		Expected Time to complete
1	Lab 7: Error Debugging with AI: Systematic approaches to finding and fixing bugs Lab Objectives: <ul style="list-style-type: none"> To identify and correct syntax, logic, and runtime errors in Python programs using AI tools. 		Week4 – Wednesday

- To understand common programming bugs and AI-assisted debugging suggestions.
- To evaluate how AI explains, detects, and fixes different types of coding errors.
- To build confidence in using AI to perform structured debugging practices.

Lab Outcomes (Los):

After completing this lab, students will be able to:

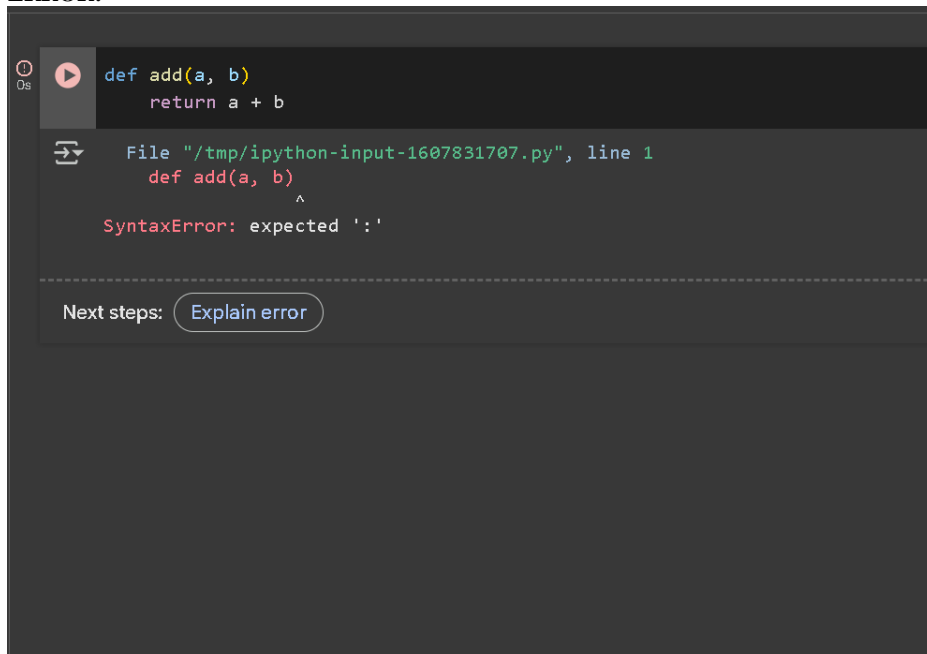
- Use AI tools to detect and correct syntax, logic, and runtime errors.
- Interpret AI-suggested bug fixes and explanations.
- Apply systematic debugging strategies supported by AI-generated insights.
- Refactor buggy code using responsible and reliable programming patterns.

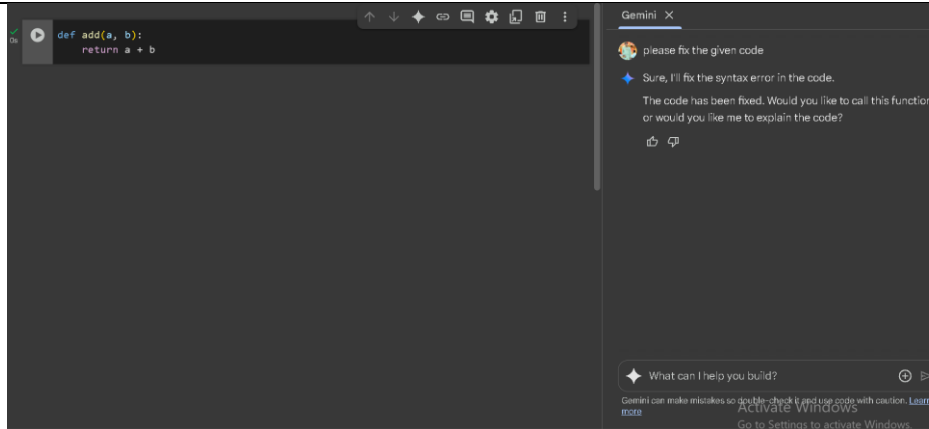
Task Description#1

- Paste a function with a missing colon (add(a, b)), and let AI fix the syntax error.

```
python

def add(a, b)
    return a + b
```

ERROR:**SOLUTION :**



PROMPT: Please fix the code

Expected Output#1

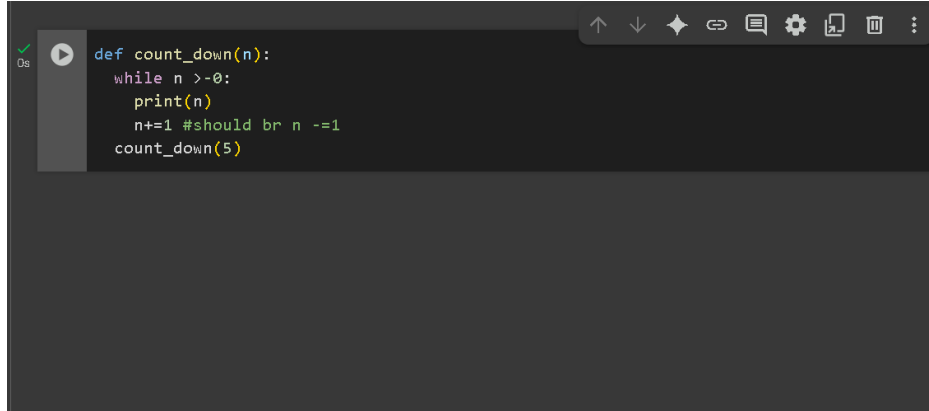
- Corrected function with syntax fix

Task Description#2 (Loops)

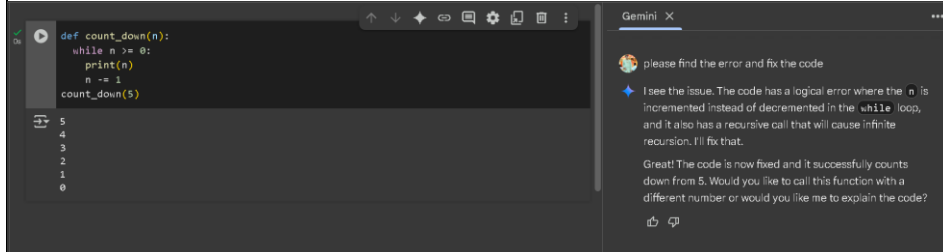
- Identify and fix a logic error in a loop that causes infinite iteration.

```
python  
  
def count_down(n):  
    while n >= 0:  
        print(n)  
        n += 1 # Should be n -= 1
```

Error:



SOLUTION:



Prompt: please find the error and fix the code

Expected Output#2

- AI fixes increment/decrement error

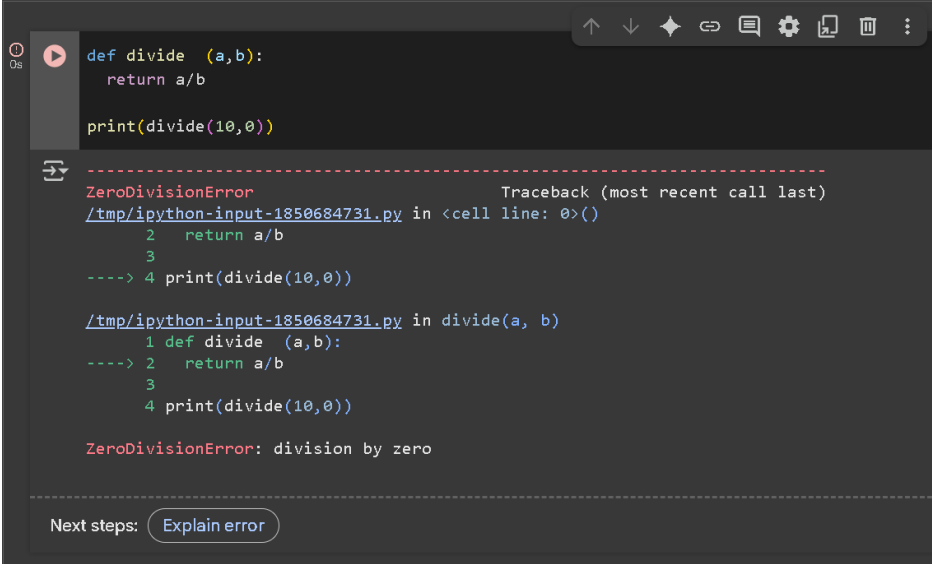
Task Description#3

- Debug a runtime error caused by division by zero. Let AI insert try-except.

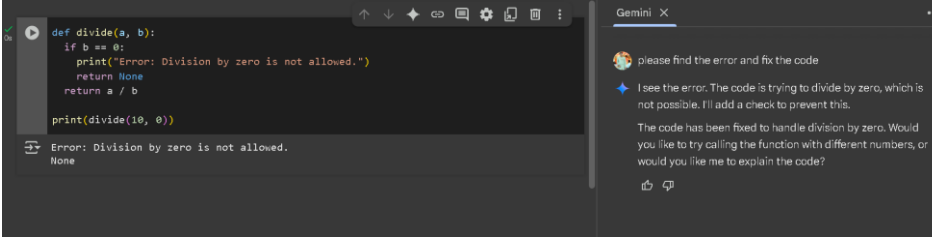
```
# Debug the following code
def divide(a, b):
    return a / b

print(divide(10, 0))
```

Error:

A screenshot of a Jupyter Notebook interface. The top part shows the code: `def divide (a,b):`, `return a/b`, and `print(divide(10,0))`. Below the code, a red error message is displayed: `ZeroDivisionError` with a traceback. The traceback shows the error occurred in `/tmp/ipython-input-1850684731.py` at line 0, and then in the `divide` function at line 2. The error message is `ZeroDivisionError: division by zero`. At the bottom, there is a 'Next steps:' section with a button labeled 'Explain error'.

Solution:

A screenshot of a Jupyter Notebook interface showing the corrected code. The code is: `def divide(a, b):`, `if b == 0:`, `print("Error: Division by zero is not allowed.")`, `return None`, `return a / b`, and `print(divide(10, 0))`. Below the code, the output is `Error: Division by zero is not allowed.` and `None`. To the right of the code editor, there is a chat window titled 'Gemini: X' showing the AI's response to the prompt 'please find the error and fix the code'. The AI's response is: 'I see the error. The code is trying to divide by zero, which is not possible. I'll add a check to prevent this. The code has been fixed to handle division by zero. Would you like to try calling the function with different numbers, or would you like me to explain the code?'

Expected Output#3

- Corrected function with safe error handling

Task Description#4

- Provide a faulty class definition (missing self in parameters). Let AI fix it

```
python

class Rectangle:
    def __init__(length, width):
        self.length = length
        self.width = width
```

Expected Output#4

- Correct `__init__()` method and explanation

Error:

```
class rectangle:
    def __init__(length, width):
        self.length = length
        self.width = width
    rect = rectangle(5.3)
```

File "<tokenize>", line 5
rect = rectangle(5.3)
^
IndentationError: unindent does not match any outer indentation level

Next steps: [Explain error](#)

Prompt: please correct the code and fix the error, aslo add the area and perimeter and give an example output

Output:

```
class rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def area(self):
        return self.length * self.width

    def perimeter(self):
        return 2 * (self.length + self.width)

# Example usage:
my_rectangle = rectangle(5.3, 2.5)
print(f'Length: {my_rectangle.length}')
print(f'Width: {my_rectangle.width}')
print(f'Area: {my_rectangle.area()}')
print(f'Perimeter: {my_rectangle.perimeter()}')
```

Length: 5.3
Width: 2.5
Area: 13.25
Perimeter: 15.6

Gemini X

please correct the code and fix the error, aslo add the area and perimeter and give an example output

I apologize for the previous error. I will correct the code, fix the class definition by adding `self`, add methods to calculate the area and perimeter of the rectangle, and provide an example of how to use the class.

Working...

Task Description#5

- Access an invalid list index and use AI to resolve the Index Error.

```
python

numbers = [1, 2, 3]
print(numbers[5])
```

Error:

```
number = [1,2,3,4,5]
print(numbers[5])
```

```
-----
NameError                                Traceback (most recent call last)
/tmp/ipython-input-4032261564.py in <cell line: 0>()
      1 number = [1,2,3,4,5]
----> 2 print(numbers[5])

NameError: name 'numbers' is not defined
```

Next steps: [Explain error](#)

Prompt: Please correct the code

Output:

```
number = [1, 2, 3, 4, 5]
index_to_access = 5

if 0 <= index_to_access < len(number):
    print(number[index_to_access])
else:
    print(f"Error: Index {index_to_access} is out of bounds for the list.")
```

Error: Index 5 is out of bounds for the list.

please correct the code

I see the error. The code is trying to access an invalid index in the list, causing an `IndexError`. I will fix this by adding a check to ensure the index is valid before accessing the element.

Working...

Expected Output#5

- AI suggests checking length or using safe access logic

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

Criteria	Max Marks
Identification of bugs	0.5
Application of AI-suggested fixes	0.5
Explanation and understanding of errors	0.5
Corrected code functionality	0.5
Report structure and reflection	0.5
Total	2.5 Marks