| 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)<br>Dr. T. Sampath Kumar<br>Dr. Pramoda Patro<br>Dr. Brij Kishor Tiwari<br>Dr.J.Ravichander<br>Dr. Mohammand Ali Shaik<br>Dr. Anirodh Kumar<br>Mr. S.Naresh Kumar<br>Dr. RAJESH VELPULA<br>Mr. Kundhan Kumar<br>Ms. Ch.Rajitha<br>Mr. M Prakash<br>Mr. B.Raju<br>Intern 1 (Dharma teja)<br>Intern 2 (Sai Prasad)<br>Intern 3 (Sowmya)<br>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 | **Applicableto Batches** | |
| **AssignmentNumber:7.3**(Present assignment number)/**24**(Total number of assignments) | | | |

| Q.No. | Question | *ExpectedTime to complete* |
|---|---|---|
| 1 | Lab 7: Error Debugging with AI: Systematic approaches to finding and fixing bugs<br><br>**Lab Objectives:**<br>• 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.

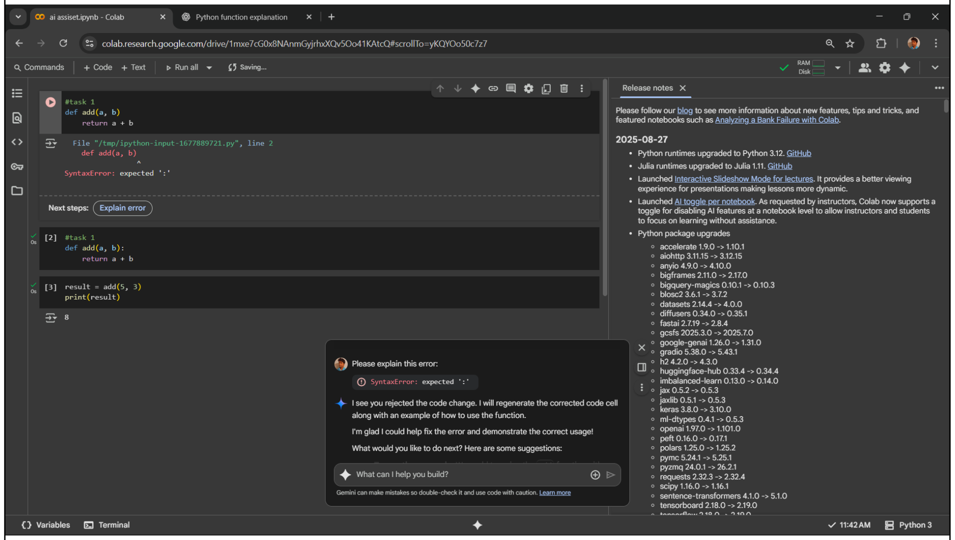
**Task Description#1**

- 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
```

**Expected Output#1**

- Corrected function with syntax fix



ai assiset.ipynb - Colab | Python function explanation

colab.research.google.com/drive/1mxe7cG0x8NAnmGyjrhxXQv5Oo41KAtcQ#scrollTo=yKQYOo50c7z7

```
#task 1
def add(a, b)
    return a + b
```

```
  File "/tmp/ipython-input-1677889721.py", line 2
    def add(a, b)
                 ^
SyntaxError: expected ':'
```

Next steps: Explain error

```
[2] #task 1
    def add(a, b):
        return a + b
```

```
[3] result = add(5, 3)
    print(result)
```

```
8
```

Please explain this error:

`SyntaxError: expected ':'`

I see you rejected the code change. I will regenerate the corrected code cell along with an example of how to use the function.

I'm glad I could help fix the error and demonstrate the correct usage!

What would you like to do next? Here are some suggestions:

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. Learn more

Release notes

Please follow our blog to see more information about new features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

2025-08-27

- Python runtimes upgraded to Python 3.12. GitHub
- Julia runtimes upgraded to Julia 1.11. GitHub
- Launched Interactive Slideshow Mode for lectures. It provides a better viewing experience for presentations making lessons more dynamic.
- Launched AI toggle per notebook. As requested by instructors, Colab now supports a toggle for disabling AI features at a notebook level to allow instructors and students to focus on learning without assistance.
- Python package upgrades
  - accelerate 1.9.0 -> 1.10.1
  - aiohttp 3.11.15 -> 3.12.15
  - anyio 4.9.0 -> 4.10.0
  - bigframes 2.11.0 -> 2.17.0
  - bigquery-magics 0.10.1 -> 0.10.3
  - blosc2 3.6.1 -> 3.7.2
  - datasets 2.14.4 -> 4.0.0
  - diffusers 0.34.0 -> 0.35.1
  - fastai 2.7.19 -> 2.8.4
  - gcsfs 2025.3.0 -> 2025.7.0
  - google-genai 1.26.0 -> 1.31.0
  - gradio 5.38.0 -> 5.43.1
  - h2 4.2.0 -> 4.3.0
  - huggingface-hub 0.33.4 -> 0.34.4
  - imbalanced-learn 0.13.0 -> 0.14.0
  - jax 0.5.2 -> 0.5.3
  - jaxlib 0.5.1 -> 0.5.3
  - keras 3.8.0 -> 3.10.0
  - ml-dtypes 0.4.1 -> 0.5.3
  - openai 1.97.0 -> 1.101.0
  - peft 0.16.0 -> 0.17.1
  - polars 1.25.0 -> 1.25.2
  - pymc 5.24.1 -> 5.25.1
  - pyzmq 24.0.1 -> 26.2.1
  - requests 2.32.3 -> 2.32.4
  - scipy 1.16.0 -> 1.16.1
  - sentence-transformers 4.1.0 -> 5.1.0
  - tensorboard 2.18.0 -> 2.19.0

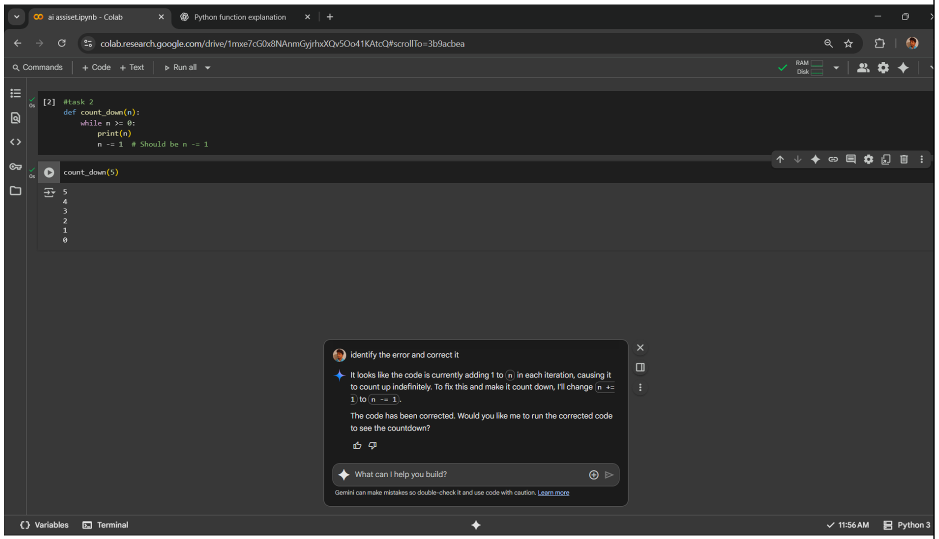
**Task Description#2 (Loops)**

- 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
```

**Expected Output#2**

- AI fixes increment/decrement error



```
[2] #task 2
    def count_down(n):
        while n >= 0:
            print(n)
            n -= 1  # Should be n -= 1
```

```
count_down(5)
```

```
5
4
3
2
1
0
```

identify the error and correct it

It looks like the code is currently adding 1 to `n` in each iteration, causing it to count up indefinitely. To fix this and make it count down, I'll change `n += 1` to `n -= 1`.

The code has been corrected. Would you like me to run the corrected code to see the countdown?

ai assiset.ipynb - Colab
Python function explanation
colab.research.google.com/drive/1mxe7cG0x8NAnmGyjrhxXQv5Oo41KAtcQ#scrollTo=3b9acbea
Commands
+ Code
+ Text
Run all
What can I help you build?
Gemini can make mistakes so double-check it and use code with caution. Learn more
Variables
Terminal
11:56 AM
Python 3

**Task Description#3**

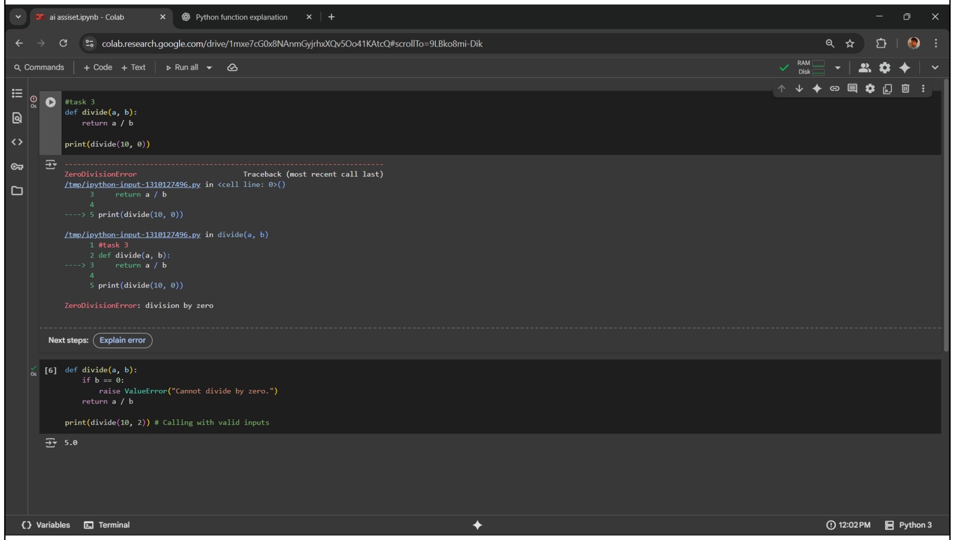
- Debug a runtime error caused by division by zero. Let AI insert try-except.

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

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

**Expected Output#3**

- Corrected function with safe error handling



```
#task 3
def divide(a, b):
    return a / b

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

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
/tmp/ipython-input-1310127496.py in <cell line: 0>()
      3     return a / b
      4 
----> 5 print(divide(10, 0))

/tmp/ipython-input-1310127496.py in divide(a, b)
      1 #task 3
      2 def divide(a, b):
----> 3     return a / b
      4 
      5 print(divide(10, 0))

ZeroDivisionError: division by zero
```

Next steps: Explain error

```
[6] def divide(a, b):
        if b == 0:
            raise ValueError("Cannot divide by zero.")
        return a / b

    print(divide(10, 2)) # Calling with valid inputs
```

```
5.0
```

ai assiset.ipynb - Colab
Python function explanation
colab.research.google.com/drive/1mxe7cG0x8NAnmGyjrhxXQv5Oo41KAtcQ#scrollTo=9LBko8mi-Dik
Commands
+ Code
+ Text
Run all
Variables
Terminal
12:02 PM
Python 3

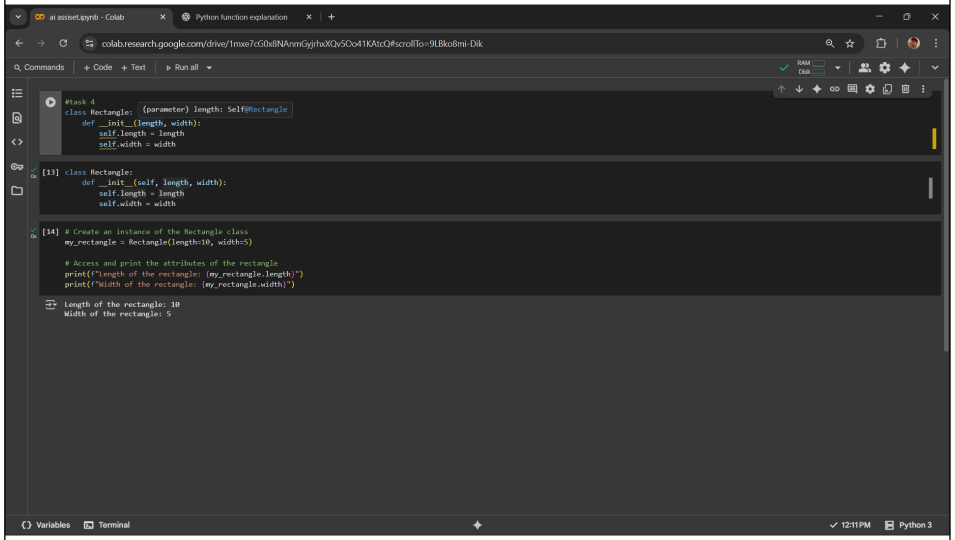
**Task Description#4**

- 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



ai assiset.ipynb - Colab | Python function explanation

colab.research.google.com/drive/1mxe7cG0x8NAnmGyjrhxXQv5Oo41KAtcQ#scrollTo=9LBko8mi-Dik

```
#task 4
class Rectangle:    (parameter) length: Self@Rectangle
    def __init__(length, width):
        self.length = length
        self.width = width
```

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

```
[14] # Create an instance of the Rectangle class
     my_rectangle = Rectangle(length=10, width=5)

     # Access and print the attributes of the rectangle
     print(f"Length of the rectangle: {my_rectangle.length}")
     print(f"Width of the rectangle: {my_rectangle.width}")
```

```
Length of the rectangle: 10
Width of the rectangle: 5
```

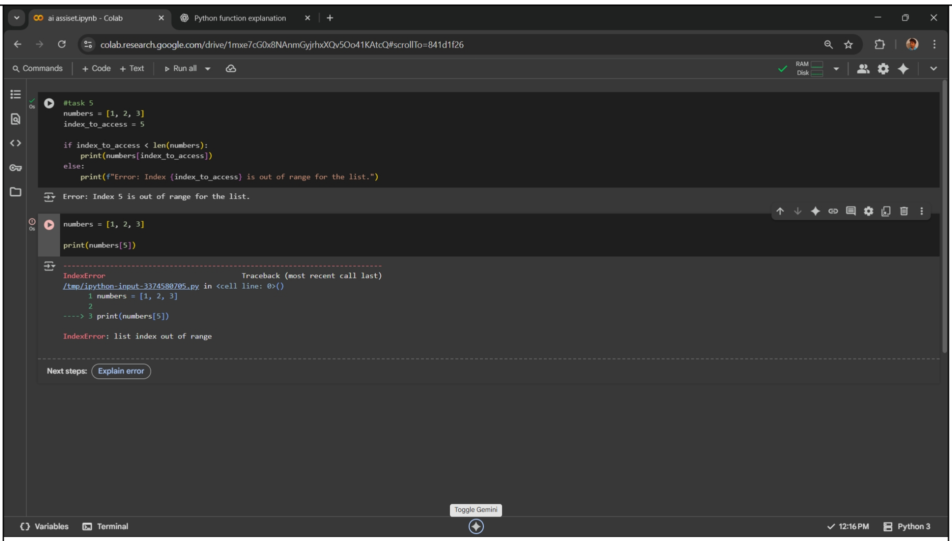
**Task Description#5**

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

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

**Expected Output#5**

- AI suggests checking length or using safe access logic



```python
#task 5
numbers = [1, 2, 3]
index_to_access = 5

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

```
Error: Index 5 is out of range for the list.
```

```python
numbers = [1, 2, 3]

print(numbers[5])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
/tmp/ipython-input-3374580705.py in <cell line: 0>()
      1 numbers = [1, 2, 3]
      2
----> 3 print(numbers[5])

IndexError: list index out of range
```

**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** |

: