| | |
|---|---|
| **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 6: AI-Based Code Completion – Classes, Loops, and Conditionals<br>**Lab Objectives:**<br>• To identify and correct syntax, logic, and runtime errors in Python programs using AI tools.<br>• To understand common programming bugs and AI-assisted debugging suggestions. | Week4 - Wednesday |

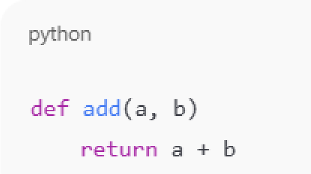
- To evaluate how AI explains, detects, and fixes different types of coding errors.
- To build confidence in using AI to perform structured debugging practices.

**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**

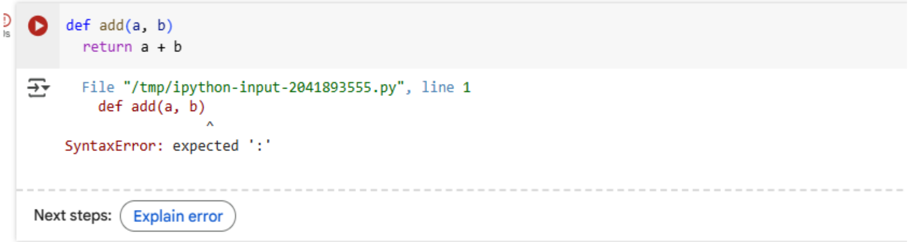
- Paste a function with a missing colon (add(a, b)), and let AI fix the syntax error.



python

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

- Video Interpolation: Predict what happened in a video between the first and the last frame.

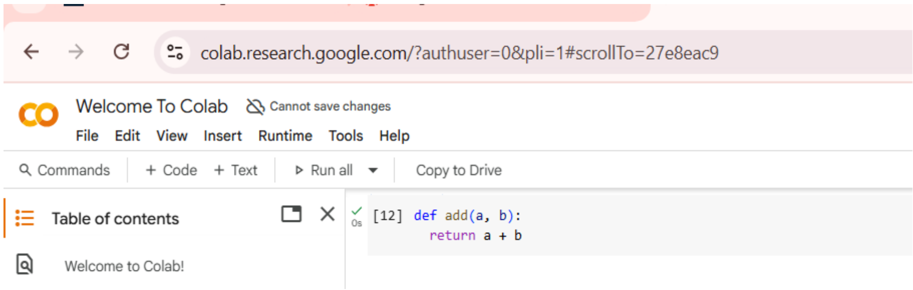


```
def add(a, b)
  return a + b
```

```
  File "/tmp/ipython-input-2041893555.py", line 1
    def add(a, b)
                 ^
SyntaxError: expected ':'
```

**Expected Output#1**

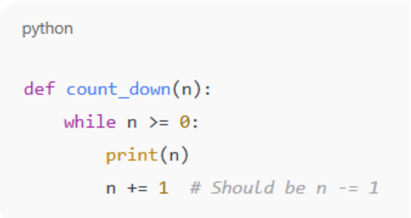
- Corrected function with syntax fix



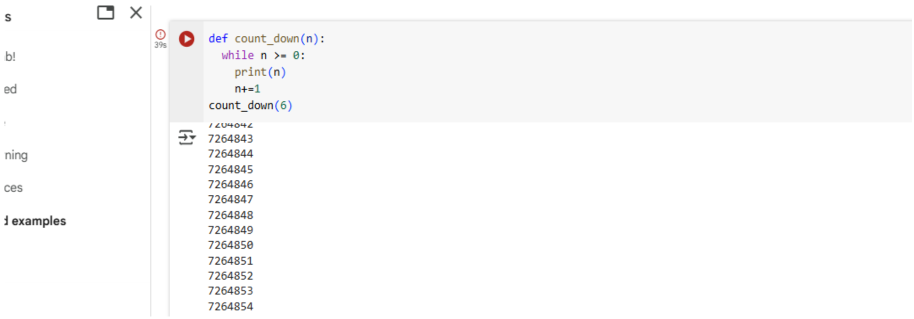
```
[12] def add(a, b):
       return a + b
```

**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
```

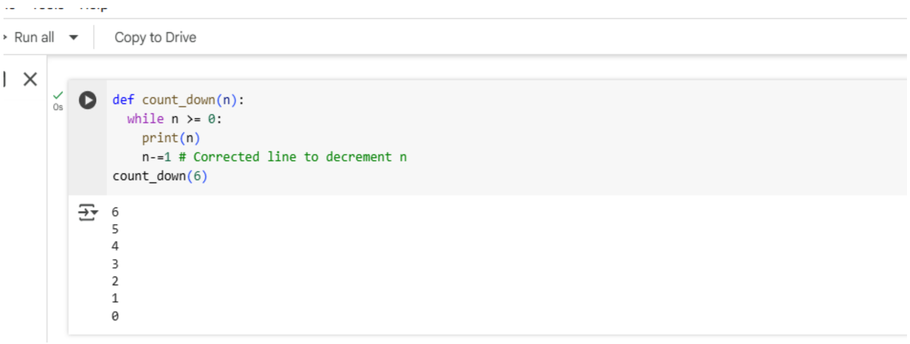


```
def count_down(n):
  while n >= 0:
    print(n)
    n+=1
count_down(6)
```

```
7264843
7264844
7264845
7264846
7264847
7264848
7264849
7264850
7264851
7264852
7264853
7264854
```

### Expected Output#2

- AI fixes increment/decrement error

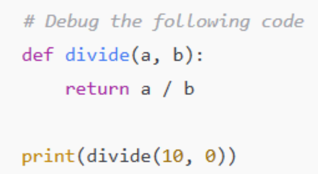


```
def count_down(n):
  while n >= 0:
    print(n)
    n-=1 # Corrected line to decrement n
count_down(6)
```

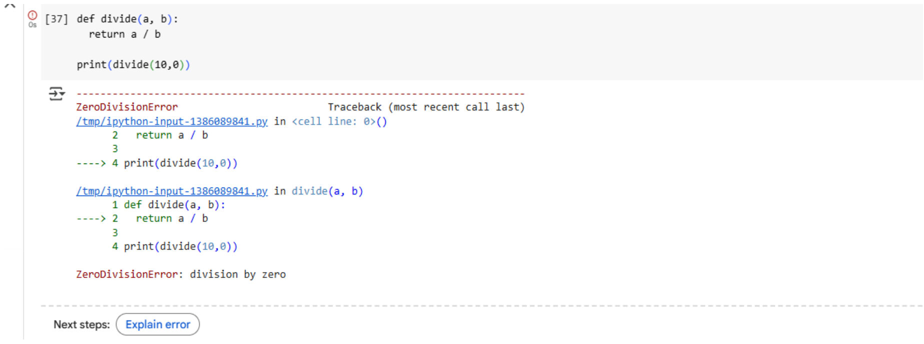
```
6
5
4
3
2
1
0
```

### Task Description#3

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



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

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

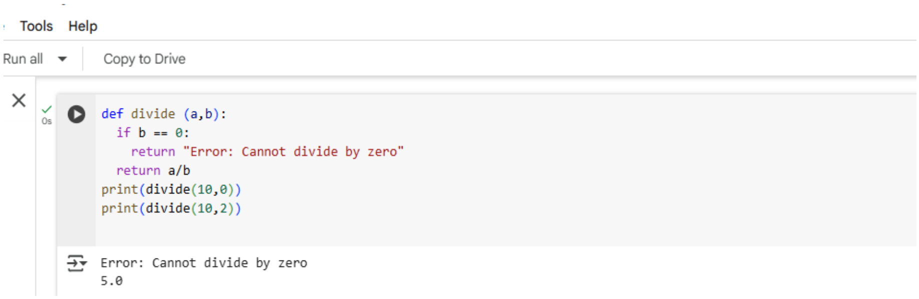


```
[37] def divide(a, b):
       return a / b

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

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
/tmp/ipython-input-1386089841.py in <cell line: 0>()
      2   return a / b
      3
----> 4 print(divide(10,0))

/tmp/ipython-input-1386089841.py in divide(a, b)
      1 def divide(a, b):
----> 2   return a / b
      3
      4 print(divide(10,0))

ZeroDivisionError: division by zero
```

**Expected Output#3**

- Corrected function with safe error handling

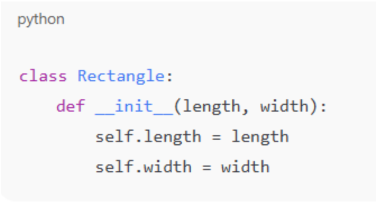


```
def divide (a,b):
  if b == 0:
    return "Error: Cannot divide by zero"
  return a/b
print(divide(10,0))
print(divide(10,2))
```

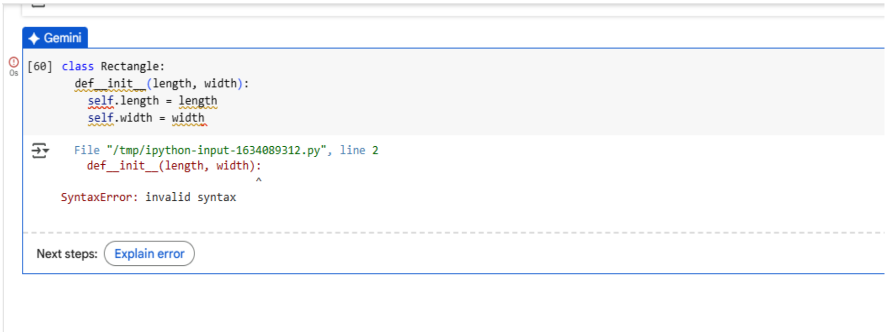
```
Error: Cannot divide by zero
5.0
```

**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
```

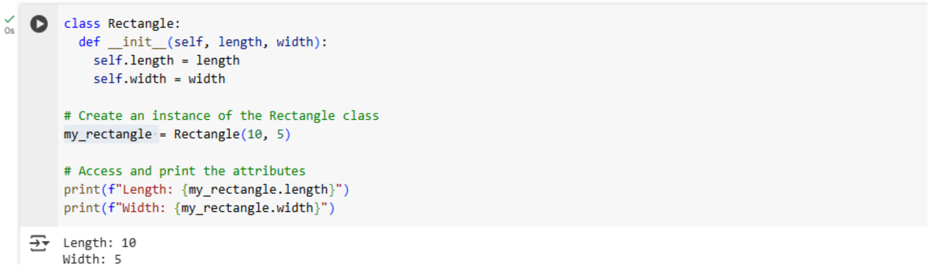


```
[60] class Rectangle:
       def__init__(length, width):
         self.length = length
         self.width = width
```

```
  File "/tmp/ipython-input-1634089312.py", line 2
    def__init__(length, width):
                              ^
SyntaxError: invalid syntax
```

**Expected Output#4**

- Correct _init () method and explanation

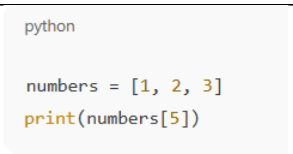


```
class Rectangle:
  def __init__(self, length, width):
    self.length = length
    self.width = width

# Create an instance of the Rectangle class
my_rectangle = Rectangle(10, 5)

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

```
Length: 10
Width: 5
```

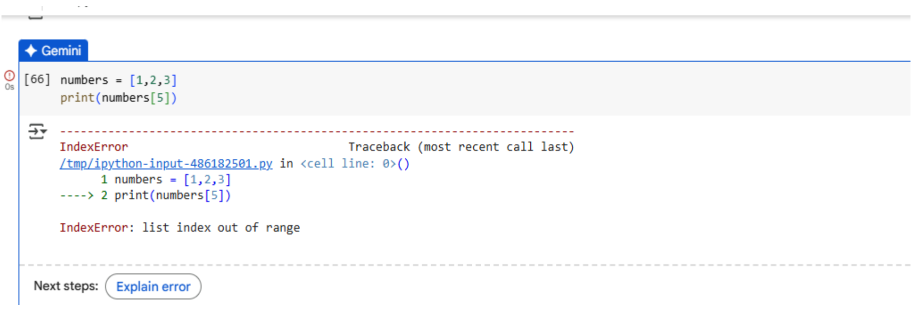
**Task Description#5**

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



python

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

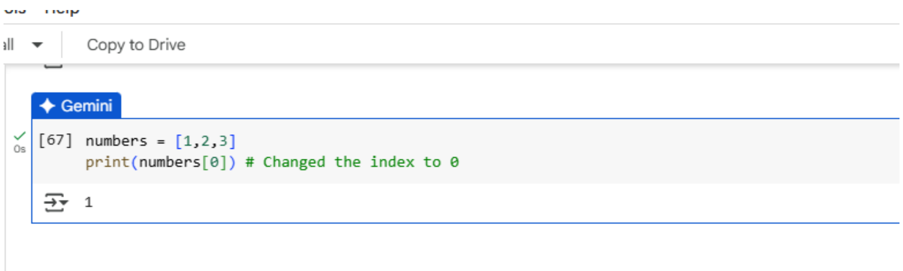


```
[66] numbers = [1,2,3]
     print(numbers[5])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
/tmp/ipython-input-486182501.py in <cell line: 0>()
      1 numbers = [1,2,3]
----> 2 print(numbers[5])

IndexError: list index out of range
```

**Expected Output#5**

- AI suggests checking length or using safe access logic



```
[67] numbers = [1,2,3]
     print(numbers[0]) # Changed the index to 0
```

```
1
```

**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** |