

<b>SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE</b>		<b>DEPARTMENT OF COMPUTER SCIENCE ENGINEERING</b>	
<b>ProgramName:</b> B. Tech		<b>Assignment Type: Lab</b>	<b>AcademicYear:</b> 2025-2026
<b>CourseCoordinatorName</b>		Venkataramana Veeramsetty	
<b>Instructor(s)Name</b>		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)			
<b>CourseCode</b>	24CS002PC215	<b>CourseTitle</b>	AI Assisted Coding
<b>Year/Sem</b>	II/I	<b>Regulation</b>	R24
<b>Date and Day of Assignment</b>	Week4 - Wednesday	<b>Time(s)</b>	
<b>Duration</b>	2 Hours	<b>Applicable to Batches</b>	
<b>AssignmentNumber:</b> 7.3(Present assignment number)/24(Total number of assignments)			
<b>Q.No.</b>	<b>Question</b>	<b>Expected Time to complete</b>	
1	Lab 7: Error Debugging with AI: Systematic approaches to finding and fixing bugs  <b>Lab Objectives:</b> <ul style="list-style-type: none"> <li>To identify and correct syntax, logic, and runtime errors in Python programs using AI tools.</li> </ul>	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

```
Ass1.py > ...
1 #Wrong input:
2 #def add(a, b). # NO colon at the end of function definition
3 | # return a + b
4 #Corrected code:
5
6 def add(a, b):
7     return a + b
8 #Expected output: 5
9 print(add(2, 3))
10
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
/usr/bin/env /usr/local/bin/python3 /Users/brungisrikar/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-x64/bu
/Users/brungisrikar/Desktop/WTMP/Ass1.py
brungisrikar@Brungis-MacBook-Pro WTMP % /usr/bin/env /usr/local/bin/python3 /Users/brungisrikar/.vscode/extensions/m
/adaptor/../../debugpy/launcher 56848 -- /Users/brungisrikar/Desktop/WTMP/Ass1.py
5
brungisrikar@Brungis-MacBook-Pro WTMP %
```

#### 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

```
Ass1.py > ...
1  #def count_down(n):
2  #   while n >= 0:
3  #       print(n)
4  #       n += 1 # This increases n, so the condition never fails
5
6  #Corrected code:
7
8  def count_down(n):
9      while n >= 0:
10         print(n)
11         n -= 1 # Decreases n, allowing the loop to eventually exit
12 # Example usage:
13 count_down(5)
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
/usr/bin/env /usr/local/bin/python3 /Users/brungisrikar/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-x64/bundled/libs/debugpy/ad
/Users/brungisrikar/Desktop/WTMP/Ass1.py
brungisrikar@Brungis-MacBook-Pro WTMP % /usr/bin/env /usr/local/bin/python3 /Users/brungisrikar/.vscode/extensions/ms-python.debugpy-2025
/adapters/.../debugpy/launcher 56901 -- /Users/brungisrikar/Desktop/WTMP/Ass1.py
5
4
3
2
1
0
brungisrikar@Brungis-MacBook-Pro WTMP %
```

### 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

```
Ass1.py > ...
1  #def divide(a, b):
2  #   return a / b
3  #print(divide(10, 0)) # Raises ZeroDivisionError
4
5
6  #Corrected code:
7  def divide(a, b):
8      try:
9          return a / b
10     except ZeroDivisionError:
11         return "Error: Cannot divide by zero"
12
13 print(divide(10, 0)) # Outputs: Error: Cannot divide by zero
14
15
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
/usr/bin/env /usr/local/bin/python3 /Users/brungisrikar/.vscode/extensions/ms-python.debugpy-2025
/Users/brungisrikar/Desktop/WTMP/Ass1.py
brungisrikar@Brungis-MacBook-Pro WTMP % /usr/bin/env /usr/local/bin/python3 /Users/brungisrikar/.
/adapters/.../debugpy/launcher 56937 -- /Users/brungisrikar/Desktop/WTMP/Ass1.py
Error: Cannot divide by zero
brungisrikar@Brungis-MacBook-Pro WTMP %
```

### 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

```
Ass1.py > ...
1  #class Rectangle:
2  #   def __init__(length, width): #   Missing 'self'
3  #       self.length = length
4  #       self.width = width
5
6
7
8  #Corrected code:
9  class Rectangle:
10     def __init__(self, length, width): #   'self' added
11         self.length = length
12         self.width = width
13 #output :
14 rect = Rectangle(10, 5)
15 print(rect.length, rect.width) # Output: 10 5
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
/usr/bin/env /usr/local/bin/python3 /Users/brungisrikar/.vscode/extensions/ms-python.de
/Users/brungisrikar/Desktop/WTMP/Ass1.py
brungisrikar@Brungis-MacBook-Pro WTMP % /usr/bin/env /usr/local/bin/python3 /Users/brun
/adaptor/../../debugpy/launcher 56993 -- /Users/brungisrikar/Desktop/WTMP/Ass1.py
10 5
brungisrikar@Brungis-MacBook-Pro WTMP %
```

#### 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

```
Ass1.py > ...
1  #numbers = [1, 2, 3]
2  #print(numbers[5]) # IndexError: list index out of range
3
4
5
6
7  #Corrected code:
8  numbers = [1, 2, 3] # Define the list first
9
10 index = 5
11 if index < len(numbers):
12     print(numbers[index])
13 else:
14     print(f"Index {index} is out of range. Max valid index is {len(numbers) - 1}.")
15
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
/usr/bin/env /usr/local/bin/python3 /Users/brungisrikar/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin
/Users/brungisrikar/Desktop/WTMP/Ass1.py
brungisrikar@Brungis-MacBook-Pro WTMP % /usr/bin/env /usr/local/bin/python3 /Users/brungisrikar/.vscode/exter
/adapiter/../../debugpy/launcher 57062 -- /Users/brungisrikar/Desktop/WTMP/Ass1.py
Index 5 is out of range. Max valid index is 2.
brungisrikar@Brungis-MacBook-Pro WTMP %
```

**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

: