NAME:Kruthankiran H.NO:2303A51404 BATCH:26

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		**Assignment Type: Lab**	**Academic Year:**2025-2026
Course Coordinator Name	Dr. Rishabh Mittal		
Instructor(s) Name	Mr. S Naresh Kumar Ms. B. Swathi Dr. Sasanko Shekhar Gantayat Mr. Md Sallauddin Dr. Mathivanan Mr. Y Srikanth Ms. N Shilpa Dr. Rishabh Mittal (Coordinator) Dr. R. Prashant Kumar Mr. Ankushavali MD Mr. B Viswanath Ms. Sujitha Reddy Ms. A. Anitha Ms. M.Madhuri Ms. Katherashala Swetha Ms. Velpula sumalatha Mr. Bingi Raju		
Course Code	23CS002PC304	**Course Title**	AI Assisted Coding
Year/Sem	III/II	**Regulation**	R23
Date and Day of Assignment	**Week4 – Wednesday**	**Time(s)**	23CSBTB01 To 23CSBTB52
Duration	2 Hours	**Applicable to Batches**	All batches
AssignmentNumber:7.3(Present assignment number)/**24**(Total number of assignments)			

Q.No.	Question	*Expected Time to complete*
1		Week4 - Wednesday

	Lab 7: Error Debugging with AI: Systematic approaches to finding and fixing bugs	
	Lab Objectives • To identify and correct syntax, logic, and runtime errors in Python programs using AI tools • 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 for 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 using AI-generated insights • Refactor buggy code using reliable programming patterns	

	Task 1: Fixing Syntax Errors
	
	Scenario
	You are reviewing a Python program where a basic function definition contains a syntax error.
	python
	`def add(a, b)`
	`    return a + b`
	
	Requirements
	• Provide a Python function add(a, b) with a **missing colon**
	• Use an AI tool to detect the syntax error
	• Allow AI to correct the function definition
	• Observe how AI explains the syntax issue
	
	Expected Output
	• Corrected function with proper syntax
	• Syntax error resolved successfully
	• AI-generated explanation of the fix	

AI 7.3.py > ...

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

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POR

```
PS C:\Users\kruth\OneDrive\Desktop\java>  & '
-python.debugpy-2025.18.0-win32-x64\bundled\l
8
PS C:\Users\kruth\OneDrive\Desktop\java>
```

Task 2: Debugging Logic Errors in Loops

Scenario
You are debugging a loop that runs infinitely due to a logical mistake.

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

Requirements

• Provide a loop with an **increment or decrement error**
• Use AI to identify the cause of infinite iteration
• Let AI fix the loop logic
• Analyze the corrected loop behavior

Expected Output
• Infinite loop issue resolved
• Correct increment/decrement logic applied
• AI explanation of the logic error

AI 7.3(i).py > ...

```
1  count = 0
2
3  while count < 10:
4      print(f"Count: {count}")
5      count += 1
6
7  print("Loop completed")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\kruth\OneDrive\Desktop\java>  & 'c:\Users\kr
-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy
8
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Use
3.11.exe' 'c:\Users\kruth\.vscode\extensions\ms-python.d
ve\Desktop\java\AI 7.3(i).py'
Count: 0
Count: 1
Count: 2
Count: 3
Count: 4
Count: 5
Count: 6
Count: 7
Count: 8
Count: 9
Loop completed
PS C:\Users\kruth\OneDrive\Desktop\java>
```

Task 3: Handling Runtime Errors (Division by Zero)

Scenario
A Python function crashes during execution due to a division by zero error.

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

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

Requirements
• Provide a function that performs division without validation
• Use AI to identify the runtime error
• Let AI add try-except blocks for safe execution
• Review AI's error-handling approach

Expected Output
• Function executes safely without crashing
• Division by zero handled using try-except
• Clear AI-generated explanation of runtime error handling

AI 7.3(ii).py > ...

```python
def divide(a, b):
    try:
        result = a / b
        return result
    except ZeroDivisionError:
        print("Error: Cannot divide by zero!")
        return None

print(divide(10, 0))
print(divide(20, 4))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Users\kruth
3.11.exe' 'c:\Users\kruth\.vscode\extensions\ms-python.debugpy-2
ve\Desktop\java\AI 7.3(i).py'
Count: 7
Count: 8
Count: 9
Loop completed
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Users\kruth
3.11.exe' 'c:\Users\kruth\.vscode\extensions\ms-python.debugpy-2
ve\Desktop\java\AI 7.3(ii).py'
Error: Cannot divide by zero!
None
5.0
PS C:\Users\kruth\OneDrive\Desktop\java>
```

Task 4: Debugging Class Definition Errors

Scenario
You are given a faulty Python class where the constructor is incorrectly defined.

	python
	
	class Rectangle:
	def __init__(length, width):
	self.length = length
	self.width = width
	
	Requirements
	• Provide a class definition with **missing self-parameter**
	• Use AI to identify the issue in the __init__() method
	• Allow AI to correct the class definition
	• Understand why self is required
	
	Expected Output
	• Corrected __init__() method
	• Proper use of self in class definition
	• AI explanation of object-oriented error	

AI 7.3(iii).py > Person > __init__

```
1   class Person:
2       def __init__(name, age):
3           name.name = name
4           name.age = age
5
6   class Person:
7       def __init__(self, name, age):
8           self.name = name
9           self.age = age
10
11  p = Person("Alice", 30)
12  print(p.name, p.age)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Users\kr
3.11.exe' 'c:\Users\kruth\.vscode\extensions\ms-python.debugp
ve\Desktop\java\AI 7.3(i).py'
Count: 9
Loop completed
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Users\kr
3.11.exe' 'c:\Users\kruth\.vscode\extensions\ms-python.debugp
ve\Desktop\java\AI 7.3(ii).py'
Error: Cannot divide by zero!
None
5.0
PS C:\Users\kruth\OneDrive\Desktop\java> ^C
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Users\kr
3.11.exe' 'c:\Users\kruth\.vscode\extensions\ms-python.debugp
ve\Desktop\java\AI 7.3(iii).py'
Alice 30
```

Task 5: Resolving Index Errors in Lists

Scenario
A program crashes when accessing an invalid index in a list.

```
python

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

Requirements

• Provide code that accesses an **out-of-range list index**
• Use AI to identify the Index Error
• Let AI suggest safe access methods
• Apply bounds checking or exception handling

Expected Output

• Index error resolved
• Safe list access logic implemented
• AI suggestion using length checks or exception handling

AI 7.3(iv).py > ...

```python
numbers = [10, 20, 30]
index = 5
if 0 <= index < len(numbers):
    print(numbers[index])
else:
    print(f"Index {index} out of range; returning default -> None")
    print(None)

try:
    print(numbers[5])
except IndexError:
    print("IndexError caught: index out of range; handling gracefully")
    print(None)

def safe_get(lst, idx, default=None):
    if -len(lst) <= idx < len(lst):
        return lst[idx]
    return default

print(safe_get(numbers, 5))
print(safe_get(numbers, -1))

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
3.11.exe' 'c:\Users\kruth\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x
5.0
PS C:\Users\kruth\OneDrive\Desktop\java> ^C
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Users\kruth\OneDrive\Deskto
3.11.exe' 'c:\Users\kruth\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x
ve\Desktop\java\AI 7.3(iii).py'
Alice 30
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Users\kruth\OneDrive\Deskto
3.11.exe' 'c:\Users\kruth\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x
ve\Desktop\java\AI 7.3(iv).py'
Index 5 out of range; returning default -> None
None
IndexError caught: index out of range; handling gracefully
None
None
30
PS C:\Users\kruth\OneDrive\Desktop\java>
```

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