

Name:P Pranay Kumar H.No:2303A51829 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		<table border="1"> <tr><td>Mr. S Naresh Kumar</td></tr> <tr><td>Ms. B. Swathi</td></tr> <tr><td>Dr. Sasanko Shekhar Gantayat</td></tr> <tr><td>Mr. Md Sallauddin</td></tr> <tr><td>Dr. Mathivanan</td></tr> <tr><td>Mr. Y Srikanth</td></tr> <tr><td>Ms. N Shilpa</td></tr> <tr><td>Dr. Rishabh Mittal (Coordinator)</td></tr> <tr><td>Dr. R. Prashant Kumar</td></tr> <tr><td>Mr. Ankushavali MD</td></tr> <tr><td>Mr. B Viswanath</td></tr> <tr><td>Ms. Sujitha Reddy</td></tr> <tr><td>Ms. A. Anitha</td></tr> <tr><td>Ms. M.Madhuri</td></tr> <tr><td>Ms. Katherashala Swetha</td></tr> <tr><td>Ms. Velpula sumalatha</td></tr> <tr><td>Mr. Bingi Raju</td></tr> </table>		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
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	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																	

<ul style="list-style-type: none">• 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 <hr/> Lab Outcomes (LOs) <p>After completing this lab, students will be able to:</p> <ul style="list-style-type: none">• 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 <hr/> Task 1: Fixing Syntax Errors <p>Scenario You are reviewing a Python program where a basic function definition contains a syntax error.</p> <pre>python def add(a, b) return a + b</pre> <p>Requirements</p> <ul style="list-style-type: none">• 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 <p>Expected Output</p> <ul style="list-style-type: none">• Corrected function with proper syntax• Syntax error resolved successfully• AI-generated explanation of the fix	
--	--

```
... AAC A 7.3.py ●
AAC A 7.3.py > ...
1 def add(a, b):
2     return a + b
3 print([add(2, 3)])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

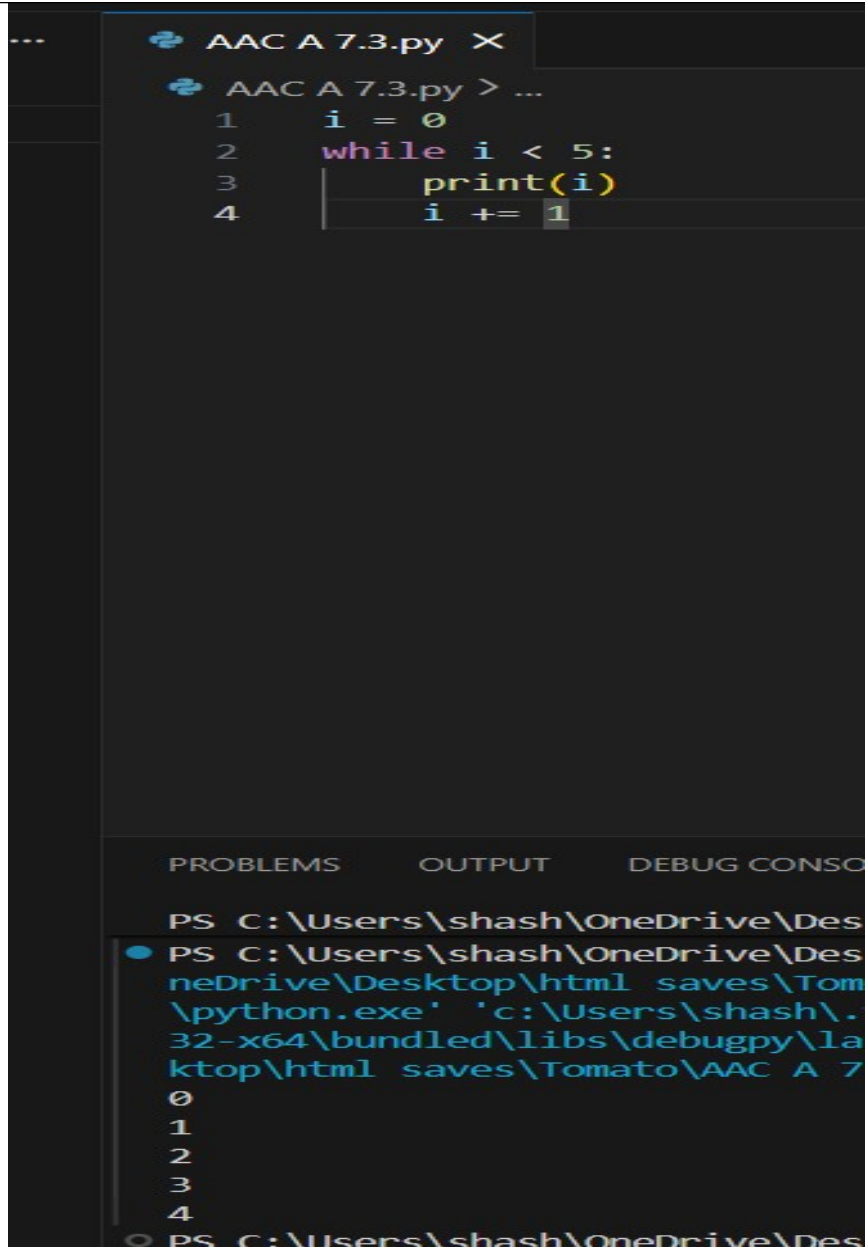
```
PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato> python lab7_debuggi
1
2
3
4
Index out of range
PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato> python "AAC A 7.3.py
● PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato>
● 5
PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato>
```

Task 2: Debugging Logic Errors in Loops

Scenario

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

	<pre>python def count_down(n): while n >= 0: print(n) n += 1 # Should be n -= 1</pre> <p>Requirements</p> <ul style="list-style-type: none">• 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 <p>Expected Output</p> <ul style="list-style-type: none">• Infinite loop issue resolved• Correct increment/decrement logic applied• AI explanation of the logic error	
--	--	--



The screenshot shows a Python IDE with a file named 'AAC A 7.3.py'. The code is a while loop that prints the value of 'i' from 0 to 4. The output window shows the numbers 0, 1, 2, 3, and 4 printed on separate lines.

```
AAC A 7.3.py X
AAC A 7.3.py > ...
1     i = 0
2     while i < 5:
3         print(i)
4         i += 1

PROBLEMS    OUTPUT    DEBUG CONSOLE

PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato\AAC A 7.3.py
PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato\AAC A 7.3.py
0
1
2
3
4
PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato\AAC A 7.3.py
```

Task 3: Handling Runtime Errors (Division by Zero)

Scenario

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

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

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

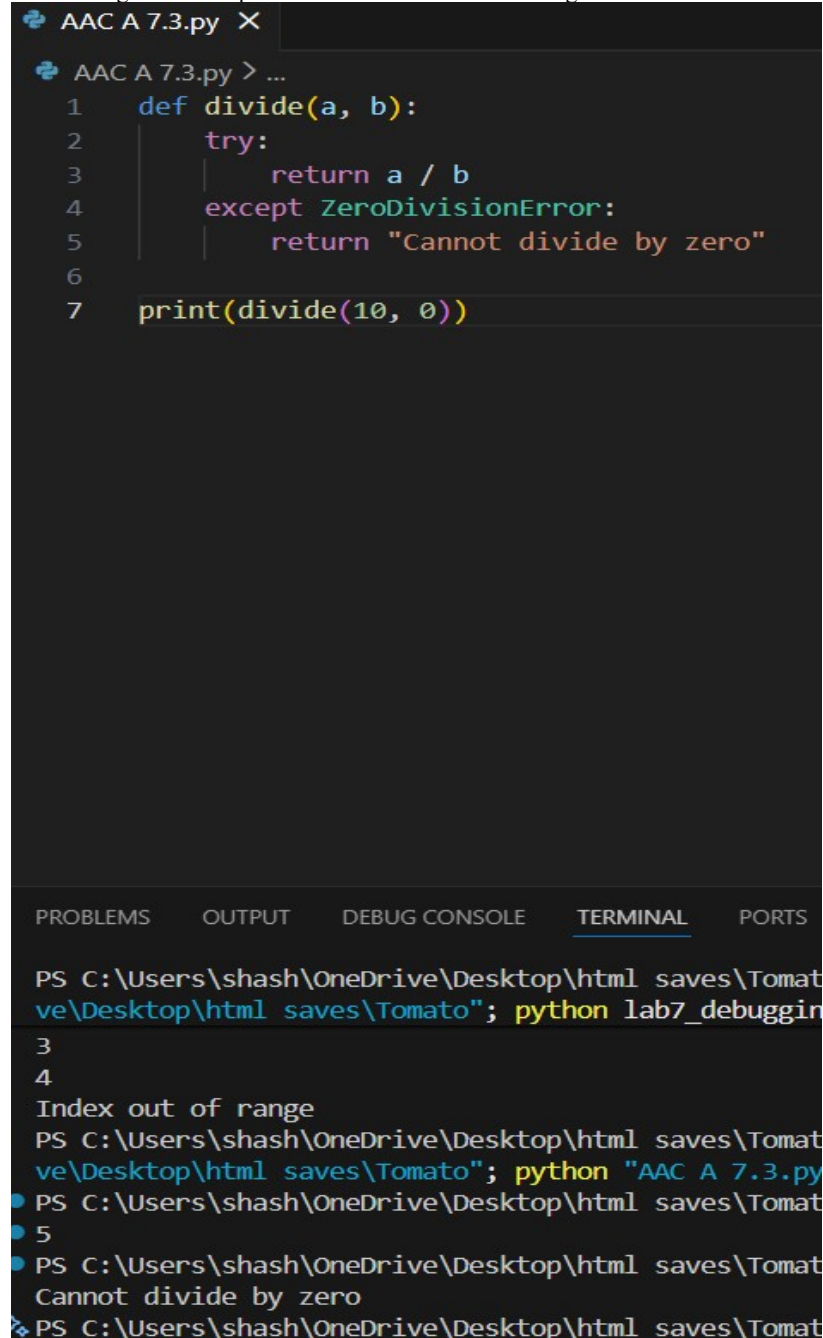
Requirements

- 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



The screenshot shows a code editor with a file named 'AAC A 7.3.py'. The code defines a 'divide' function that uses a try-except block to handle a ZeroDivisionError. Below the code, the 'TERMINAL' tab shows the command 'python lab7_debuggin' being executed, followed by an 'Index out of range' error. A subsequent command 'python "AAC A 7.3.py"' is shown, followed by the output 'Cannot divide by zero'.

```
AAC A 7.3.py X
AAC A 7.3.py > ...
1  def divide(a, b):
2      try:
3          return a / b
4      except ZeroDivisionError:
5          return "Cannot divide by zero"
6
7  print(divide(10, 0))

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

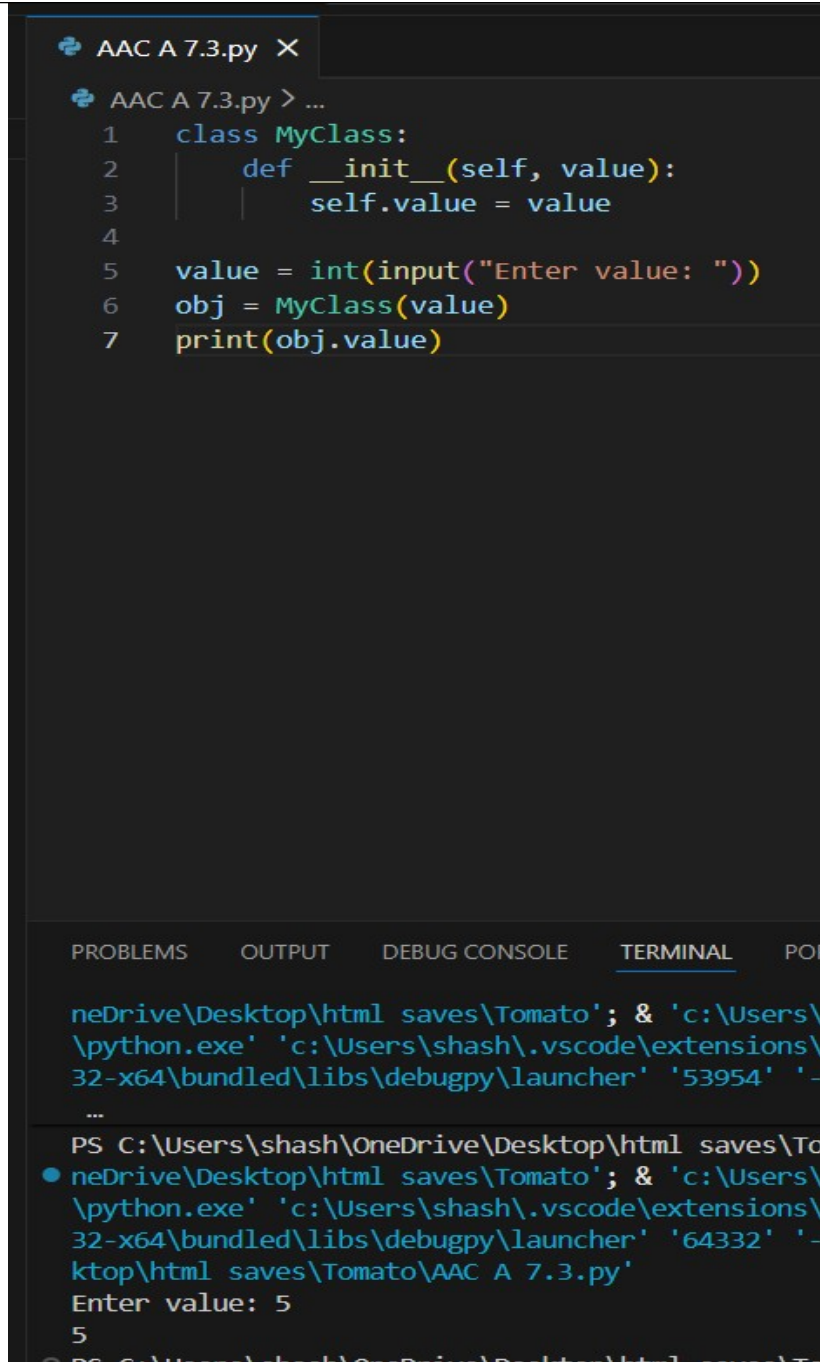
PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato\
ve\Desktop\html saves\Tomato"; python lab7_debuggin
3
4
Index out of range
PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato\
ve\Desktop\html saves\Tomato"; python "AAC A 7.3.py
• PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato
• 5
• PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato
  Cannot divide by zero
❖ PS C:\Users\shash\OneDrive\Desktop\html saves\Tomato
```

Task 4: Debugging Class Definition Errors

Scenario

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

	<div data-bbox="378 195 751 390"><pre>python class Rectangle: def __init__(length, width): self.length = length self.width = width</pre></div> <div data-bbox="378 426 919 564"><p>Requirements</p><ul style="list-style-type: none">• Provide a class definition with missing self-parameter• Use AI to identify the issue in the <code>__init__()</code> method• Allow AI to correct the class definition• Understand why self is required</div> <div data-bbox="378 590 761 701"><p>Expected Output</p><ul style="list-style-type: none">• Corrected <code>__init__()</code> method• Proper use of self in class definition• AI explanation of object-oriented error</div>	
--	--	--



The screenshot shows a VS Code editor with a file named 'AAC A 7.3.py'. The code defines a class 'MyClass' with an '__init__' method that sets 'self.value' to the provided 'value'. It then prompts the user to 'Enter value: ', takes input, creates an instance of 'MyClass', and prints the 'value' attribute. The terminal at the bottom shows the command to run the script, the prompt 'Enter value: 5', and the output '5'.

```
AAC A 7.3.py X
AAC A 7.3.py > ...
1 class MyClass:
2     def __init__(self, value):
3         self.value = value
4
5 value = int(input("Enter value: "))
6 obj = MyClass(value)
7 print(obj.value)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POP
neDrive\Desktop\html saves\Tomato'; & 'c:\Users\
\python.exe' 'c:\Users\shash\.vscode\extensions\
32-x64\bundled\libs\debugpy\launcher' '53954' '-
...
PS C:\Users\shash\OneDrive\Desktop\html saves\To
neDrive\Desktop\html saves\Tomato'; & 'c:\Users\
\python.exe' 'c:\Users\shash\.vscode\extensions\
32-x64\bundled\libs\debugpy\launcher' '64332' '-
ktop\html saves\Tomato\AAC A 7.3.py'
Enter value: 5
5
PS C:\Users\shash\OneDrive\Desktop\html saves\To
```

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

```
AAC A 7.3.py > ...
1 my_list = [1, 2, 3]
2 try:
3     print(my_list[5])
4 except IndexError:
5     print("Index out of range")
```

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
neDrive\Desktop\html saves\Tomato'; & 'c:\Use
\python.exe' 'c:\Users\shash\.vscode\extensio
32-x64\bundled\libs\debugpy\launcher' '64332'
...
5
PS C:\Users\shash\OneDrive\Desktop\html saves
neDrive\Desktop\html saves\Tomato'; & 'c:\Use
\python.exe' 'c:\Users\shash\.vscode\extensio
32-x64\bundled\libs\debugpy\launcher' '64395'
ktop\html saves\Tomato\AAC A 7.3.py'
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