

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		Assignment Type: Lab	
Course Coordinator Name		Dr. Rishabh Mittal	
CourseCode	23CS002PC304	Course Title	AI Assisted Coding
Year/Sem	III/II	Regulation	R23
Date and Day of Assignment	Week4 – Monday	Batch	23CSBTB47B
Name	Mittapalli Yashaswini	Hall Ticket No	2303A54049
Assignment Number: 7.1			

Q.No.	Question	Expected Time to complete
1	<p>Lab 7: Error Debugging with AI: Systematic approaches to finding and fixing bugs</p> <p>Lab Objectives:</p> <ul style="list-style-type: none"> 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 to perform structured debugging practices. <p>Lab Outcomes (LOs):</p> <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 supported by AI-generated insights. Refactor buggy code using responsible and reliable programming patterns. 	Week4 - Monday

Task Description #1 (Syntax Errors – Missing Parentheses in Print Statement)

Task: Provide a Python snippet with a missing parenthesis in a print statement (e.g., print "Hello"). Use AI to detect and fix the syntax error.

```
# Bug: Missing parentheses in print statement
```

```
def greet():
    print "Hello, AI Debugging Lab!"
greet()
```

Requirements:

- Run the given code to observe the error.
- Apply AI suggestions to correct the syntax.
- Use at least 3 assert test cases to confirm the corrected code works.

Expected Output #1:

- Corrected code with proper syntax and AI explanation.

```
LAB-7.1.py
qodo > #TASK1:Provide a Python snippet with a missing parenthesis in a printstatement (e.g., print "Hello"). Use AI to detect and fix the syntax
      Oodo: test this function
      1
      2 def greet():
      3     print(greet())
      4     print("Hello, AI Debugging Lab!")
      5 assert greet() == "Hello, AI Debugging Lab!"
      6 assert isinstance(greet(), str)
      7 assert "AI Debugging" in greet()
      8 print("All test cases passed successfully!")

PROBLEMS ① OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER ①
PS C:\Users\bachu\OneDrive\Desktop\SRUJ-3\AI_AC\Assignment-4.1> conda activate base
PS C:\Users\bachu\OneDrive\Desktop\SRUJ-3\AI_AC\Assignment-4.1> & C:/Users/bachu/miniforge3/miniforge3/python.exe c:/Users/bachu/OneDrive/Desktop/SRUJ-3\AI_AC\Assignment-4.1/qodo/LAB-7.1.py
Hello, AI Debugging Lab!
All test cases passed successfully!
PS C:\Users\bachu\OneDrive\Desktop\SRUJ-3\AI_AC\Assignment-4.1>
```

Task Description #2 (Incorrect condition in an If Statement)

Task: Supply a function where an if-condition mistakenly uses = instead of ==. Let AI identify and fix the issue.

```
# Bug: Using assignment (=) instead of comparison (==)
```

```
def check_number(n):\n    if n = 10:\n        return "Ten"\n    else:\n        return "Not Ten"
```

Requirements:

- Ask AI to explain why this causes a bug.
 - Correct the code and verify with 3 assert test cases.

Expected Output #2:

- Corrected code using `==` with explanation and successful test execution.

The screenshot shows a Jupyter Notebook interface with the following code in a cell:

```
#qndo > ◆ LAB-7.1.py ...
8 # print("All test cases passed successfully!")
9
10 #TUTORIAL:Supply a function where an if-condition mistakenly uses == instead of ===. Let AI identify and fix the issue
Ques: How many?
11 def check_number(n):
12     if n == 10:
13         return "Ten"
14     else:
15         return "Not Ten"
16
17 print(check_number(10))
18 print(check_number(5))
19 assert check_number(10) == "Ten"
20 assert check_number(5) == "Not Ten"
21
22 print("All test cases passed successfully!")
```

The notebook also displays the following output at the bottom:

```
PS C:\Users\bachu\OneDrive\Desktop\SRUJ-3-2\AI_AC\Assignment-4.1> & C:/Users/bachu/miniforge3/miniforge3/python.exe c:/Users/bachu/OneDrive/Desktop/SRUJ-3-2/AI_AC/Assignment-4.1/qndo/LAB-7.1.py
True
Not Ten
All test cases passed successfully!
PS C:\Users\bachu\OneDrive\Desktop\SRUJ-3-2\AI_AC\Assignment-4.1>
```

Task Description #3 (Runtime Error – File Not Found)

Task: Provide code that attempts to open a non-existent file and crashes. Use AI to apply safe error handling.

Bug: Program crashes if file is missing

```
def read_file(filename):
    with open(filename, 'r') as f:
        return f.read()
print(read_file("nonexistent.txt"))
```

Requirements:

- Implement a try-except block suggested by AI.

- Add a user-friendly error message.
- Test with at least 3 scenarios: file exists, file missing, invalid path.

Expected Output #3:

- Safe file handling with exception management.

```
def read_file(filename):
```

```
    try:
```

```
        with open(filename, 'r') as f:
```

```
            return f.read()
```

```
    except FileNotFoundError:
```

```
        return "File not found!"
```

```
    except Exception as e:
```

```
        return f"Error: {e}"
```

```
print(read_file("nonexistent.txt"))
```

```
#TASK: Provide code that attempts to open a non-existent file and crashes. Use AI to apply safe error handling
def read_file(filename):
    try:
        with open(filename, 'r') as f:
            return f.read()
    except FileNotFoundError:
        return "File not found!"
    except Exception as e:
        return f"Error: {e}"
print(read_file("nonexistent.txt"))

PS C:\Users\bachu\OneDrive\Desktop\SRUJ-3-2\AI_AC\Assignment-4> & C:/Users/bachu/miniforge3/miniforge3/python.exe c:/Users/bachu/OneDrive/Desktop/SRUJ-3-2\AI_AC\Assignment-4\1\qdo/LAB-7-1.py
File not found!
```

Task Description #4 (Calling a Non-Existent Method)

Task: Give a class where a non-existent method is called (e.g., `obj.undefined_method()`). Use AI to debug and fix.

```
# Bug: Calling an undefined method
```

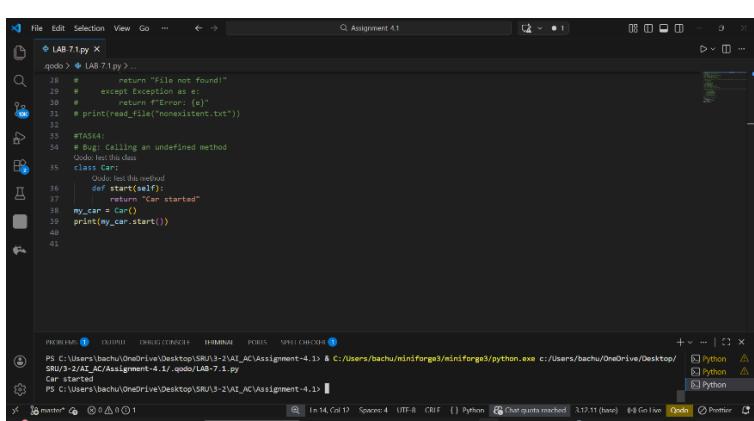
```
class Car:
```

```
    def start(self):
```

```
        return "Car started"
```

```
my_car = Car()
```

```
print(my_car.drive()) # drive() is not defined
```

Requirements:	<ul style="list-style-type: none">Students must analyze whether to define the missing method or correct the method call.Use 3 assert tests to confirm the corrected class works. <p>Expected Output #4:</p> <ul style="list-style-type: none">Corrected class with clear AI explanation.  <p>Task Description #5 (TypeError – Mixing Strings and Integers in Addition)</p> <p>Task: Provide code that adds an integer and string ("5" + 2) causing a TypeError. Use AI to resolve the bug.</p> <pre># Bug: TypeError due to mixing string and integer def add_five(value): return value + 5 print(add_five("10"))</pre> <p>Requirements:</p> <ul style="list-style-type: none">Ask AI for two solutions: type casting and string concatenation.Validate with 3 assert test cases. <p>Expected Output #5:</p> <ul style="list-style-type: none">Corrected code that runs successfully for multiple inputs.
---------------	---

The screenshot shows a code editor interface with a Python file named `LAB-7.1.py` open. The code contains several comments and a function definition:

```
qodoo > # LAB-7.1.py > ...
34 # # bug: Calling an undefined method
35 # class Car:
36 #     def start(self):
37 #         return "car started"
38 # my_car = Car()
39 # print(my_car.start())
40
41 #Task5: Provide code that adds an integer and string ("5" + 2) causing a TypeError. Use AI to resolve the bug.
42 def add_five(value):
43     return value + 5
44 print(add_five(10))
```

Below the code editor is a terminal window showing command-line output:

```
PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SPELL CHECKER
PS C:\Users\bachu\OneDrive\Desktop\SRUJ-3-2\AI_AC\Assignment-4.1> & C:/Users/bachu/miniforge3/python.exe c:/Users/bachu/OneDrive/Desktop/SRUJ-3-2\AI_AC\Assignment-4.1\qodoo/LAB-7.1.py
15
PS C:\Users\bachu\OneDrive\Desktop\SRUJ-3-2\AI_AC\Assignment-4.1>
```

The terminal also displays system status information at the bottom:

```
Moody cloudy 28°C Search 11, Col 23 Spaces: 4 UTF-8 CRLF Python Chat quota reached 3.12.11 (base) Go Live Qodoo Prettier
ENG IN 10-02-2023 23%
```