

## ASSIGNMENT-7.1

Name:M Varshith Reddy

HTNO:2303A52087

Batch:40

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()
```

The screenshot shows a code editor interface with a dark theme. A Python script is open in the editor. The code contains a function definition for 'greet' which prints a message. There is a syntax error on the third line where 'print' is followed by a variable 'h' without a closing parenthesis. The word 'print' is highlighted in blue, and 'h' is underlined in red with a small error icon. Below the code editor, there is a terminal window showing the command prompt 'PS C:\...' and the path 'OneDrive\Desktop\AI\_assisstant\_codding> & C:/Users/sravan/AppData/Local/Programs/Python/3.8.5/python.exe'. It then shows the file being run: 'File "c:/Users/sravan/Desktop/AI\_assisstant\_codding/lab07.py", line 3'. The error message 'SyntaxError: '(' was never closed' is displayed, indicating the missing parenthesis. The terminal window also shows the command prompt again at the end.

```
1
2     h="hello world"
3     print(h)
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\sravan\OneDrive\Desktop\AI_assisstant_codding> & C:/Users/sravan/AppData/Local/Programs/Python/3.8.5/python.exe
File "c:/Users/sravan/Desktop/AI_assisstant_codding/lab07.py", line 3
    print(h
)
SyntaxError: '(' was never closed
PS C:\Users\sravan\OneDrive\Desktop\AI_assisstant_codding>
```

## 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):
```

```
if n = 10:
```

```
    return "Ten"
```

```
else:
```

```
    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 code editor with a dark theme. A Python script named 'lab07.py' is open. On line 4, there is an if statement: 'if n=10:'. The '=' operator is highlighted in red, indicating a syntax error. The code editor has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. Below the code, the terminal window shows the following output:

```
PS C:\Users\sravan\OneDrive\Desktop\AI_assisstant_coding> & C:/Users/sravan/AppData/Local/Programs/Python/Python313/pyt
rs/sravan/OneDrive/Desktop/AI_assisstant_coding/lab07.py
File "C:/Users/sravan/OneDrive/Desktop/AI_assisstant_coding/lab07.py", line 4
  if n=10:
    ^
SyntaxError: invalid syntax. Maybe you meant '==' or ':' instead of '='?
PS C:\Users\sravan\OneDrive\Desktop\AI_assisstant_coding>
```

### 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.

The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for PROBLEMS (with 3 notifications), OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. Below the tabs, the code is displayed:

```
1 def read_file(filename):
2     with open(filename, 'r') as f:
3         return f.read()
4     print(read_file("nonexistent.txt"))
5 
```

The code editor highlights several lines in red, indicating syntax errors:

- Line 2: `with open(filename, 'r') as f:`
- Line 4: `print(read_file("nonexistent.txt"))`

Below the code, the terminal output shows the error message:

```
PS C:\Users\sravan\OneDrive\Desktop\AI_assisstant_coding> & C:/Users/sravan/AppData/Local/Temp/Desktop/AI_assisstant_coding/lab07.py
File "c:/Users/sravan/OneDrive/Desktop/AI_assisstant_coding/lab07.py", line 2
    with open(filename, 'r') as f:
        ^
IndentationError: expected an indented block after function definition on line 1
PS C:\Users\sravan\OneDrive\Desktop\AI_assisstant_coding>
```

#### 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):

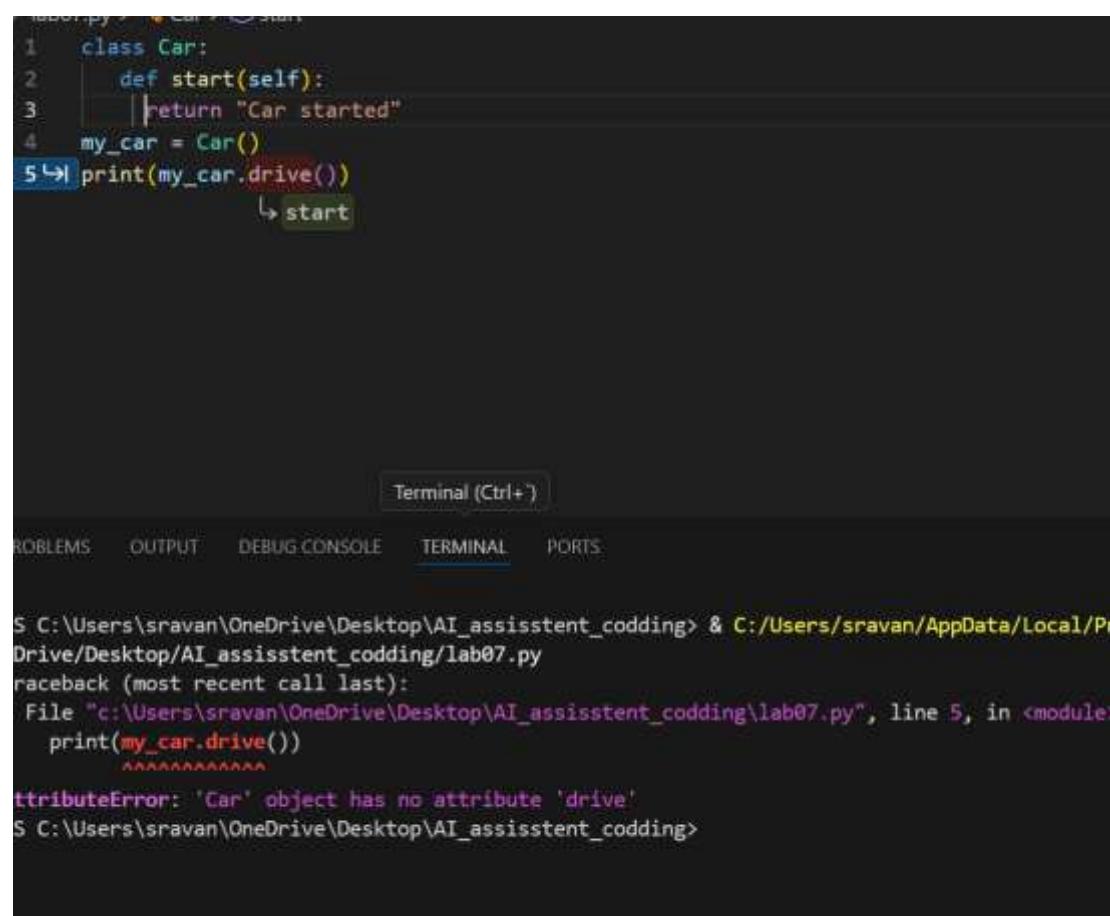
return "Car started"

my\_car = Car()

print(my\_car.drive()) # drive() is not defined

Requirements:

- Students must analyze whether to define the missing method or correct the method call.
- Use 3 assert tests to confirm the corrected class works.



```
lab07.py └── Car └── start
1  class Car:
2      def start(self):
3          return "Car started"
4  my_car = Car()
5 ↵ print(my_car.drive())
           ↵ start
```

Terminal (Ctrl+T)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
S C:\Users\srajan\OneDrive\Desktop\AI_assisstant_coding & C:/Users/srajan/AppData/Local/Programs/Desktop/AI_assisstant_coding/lab07.py
raceback (most recent call last):
  File "c:\Users\srajan\OneDrive\Desktop\AI_assisstant_coding\lab07.py", line 5, in <module>
    print(my_car.drive())
               ^
AttributeError: 'Car' object has no attribute 'drive'
S C:\Users\srajan\OneDrive\Desktop\AI_assisstant_coding>
```

Task Description #5 (TypeError – Mixing Strings and Integers in Addition)

Task: Provide code that adds an integer and string ("5" + 2) causing a TypeError. Use AI to resolve the bug.

```
# Bug: TypeError due to mixing string and integer

def add_five(value):

    return value + 5

print(add_five("10"))
```

Requirements:

- Ask AI for two solutions: type casting and string concatenation.
- Validate with 3 assert test cases.

The screenshot shows a terminal window with the following content:

```
lab07.py > add_five
1 def add_five(value):
2     return value + 5
3 print(add_five("10"))  print(add_five(10))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\sravan\OneDrive\Desktop\AI_assisstant_coding> & C:/Users/sravan/AppData/Local/Programs/Python/Python38-32/eDrive/Desktop/AI_assisstant_coding/lab07.py
Traceback (most recent call last):
  File "c:\Users\sravan\OneDrive\Desktop\AI_assisstant_coding\lab07.py", line 3, in <module>
    print(add_five("10"))
           ^^^^^^
  File "c:\Users\sravan\OneDrive\Desktop\AI_assisstant_coding\lab07.py", line 2, in add_five
    return value + 5
           ^^^^^^
TypeError: can only concatenate str (not "int") to str
%
```