

Assingment-7.1

Name: Y. Poojitha

Hall Ticket No:2303A51499

Batch-08

Lab 7: Error Debugging with AI: Systematic approaches to finding and fixing bugs

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.

```
AIAC.py 2 X
AIAC.py > ...
1  # Bug: Missing parentheses in print statement
2  def greet():
3      print "Hello, AI Debugging Lab!"
4  greet()
5
6
```

PROBLEMS 2 TERMINAL OUTPUT DEBUG CONSOLE PORTS POSTMAN CONSOLE Python Debug Console

```
PS D:\python_dsa> & 'c:\python314\python.exe' 'c:\Users\yaraV\.vscode\extensions\ms-python.python-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52029' '--'
`'
File "d:\python_dsa\AIAC.py", line 3
print "Hello, AI Debugging Lab!"
^
IndentationError: expected an indented block after function definition on line 2
PS D:\python_dsa>
```

Corrected Code

The screenshot shows the VS Code interface with the following details:

- Editor:** The file `AIAC.py` is open, showing a single line of code:

```
1 # Bug: Missing parentheses in print statement
2 def greet():
3     print("Hello, AI Debugging Lab!")
4 greet()
```
- Terminal:** The terminal shows the command `python dsa & 'c:\Python314\python.exe' 'c:/Users/yarav/.vscode/extensions/ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53869' '--' 'd:/python_dsa/AIAC.py'` being run, followed by the output "Hello, AI Debugging Lab!"
- Bottom Bar:** The tabs PROBLEMS, TERMINAL, OUTPUT, DEBUG CONSOLE, PORTS, and POSTMAN CONSOLE are visible. The TERMINAL tab is selected.

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.

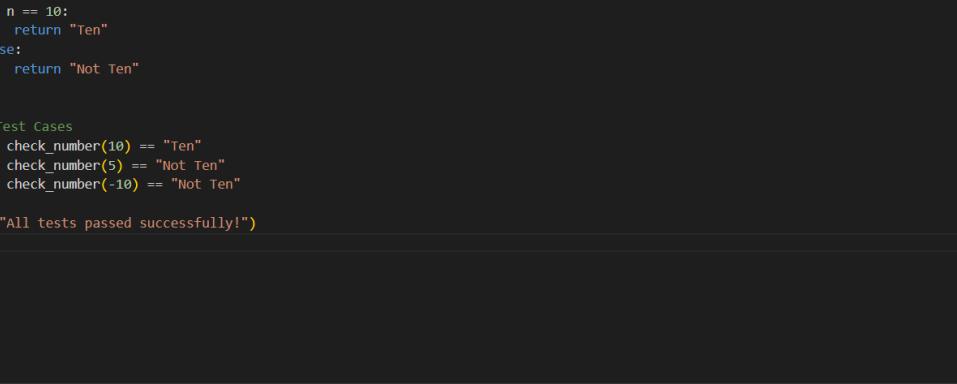
The screenshot shows the VS Code interface with the following details:

- Editor:** The file `AIAC.py` is open, showing a bug in the `check_number` function:

```
1 # Bug: Using assignment (=) instead of comparison (==)
2 def check_number(n):
3     if n = 10:
4         return "Ten"
5     else:
6         return "Not Ten"
```
- Terminal:** The terminal shows the command `python dsa & 'c:\Python314\python.exe' 'c:/Users/yarav/.vscode/extensions/ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53869' '--' 'd:/python_dsa/AIAC.py'` being run, followed by the output "Hello, AI Debugging Lab!". It then shows an error message:

```
PS D:\python_dsa> ^C
PS D:\python_dsa> PS D:\python_dsa> d:; cd 'd:/python_dsa'; & 'c:\Python314\python.exe' 'c:/Users/yarav/.vscode/extensions/ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53869' '--' 'd:/python_dsa/AIAC.py'
File "d:/python_dsa/AIAC.py", line 3
  if n = 10:
      ^^^^^^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```
- Bottom Bar:** The tabs PROBLEMS, TERMINAL, OUTPUT, DEBUG CONSOLE, PORTS, and POSTMAN CONSOLE are visible. The TERMINAL tab is selected.

Corrected Code



The screenshot shows a Python file named AIAC.py in the VS Code editor. The code defines a function check_number that returns "Ten" if the input n is 10, and "Not Ten" otherwise. It includes three test cases using assert statements and a final print statement confirming all tests passed.

```
AIAC.py
1 # Fixed Version: Using comparison operator (==)
2 def check_number(n):
3     if n == 10:
4         return "Ten"
5     else:
6         return "Not Ten"
7
8
9 #  Test Cases
10 assert check_number(10) == "Ten"
11 assert check_number(5) == "Not Ten"
12 assert check_number(-10) == "Not Ten"
13
14 print("All tests passed successfully!")
15
16
17
```

TERMINAL

```
PS D:\python_dsa> & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '57860'
y'
All tests passed successfully!
PS D:\python_dsa>
```

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.

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a file tree with 'AIAC.py' as the active file.
- Code Editor:** Displays the Python code for 'AIAC.py'. The code defines a function 'read_file' that reads a file by its name. It includes a comment about a bug where the program crashes if the file is missing. A call to 'read_file' with the argument 'nonexistent.txt' is shown at the bottom.
- Terminal:** Shows the command-line interface with the following history:
 - 'All tests passed successfully!'
 - 'PS D:\python_dsa> ^C'
 - 'PS D:\python_dsa> PS D:\python_dsa> PS D:\python_dsa> d;; cd 'd:\python_dsa'; & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy_vendored_pydevd.py''
 - 'Traceback (most recent call last):'
 - 'File "d:\python_dsa\AIAC.py", line 6, in <module>' followed by the full stack trace.
- Bottom Status Bar:** Shows 'Python Debug Console'.

Safe Version with Exception Handling

```
AIAC.py > ...
1 def read_file(filename):
2     try:
3         with open(filename, 'r') as f:
4             return f.read()
5     except FileNotFoundError:
6         return "Error: File not found."
7     except OSError:
8         return "Error: Invalid file path or unable to access file."
9
10 # -----
11 # ✅ Test Scenarios
12 # -----
13
14 # 1 File Exists
15 with open("testfile.txt", "w") as f:
16     f.write("Sample content")
17
18 assert read_file("testfile.txt") == "Sample content"
19
20 # 2 File Missing
21 assert read_file("nonexistent.txt") == "Error: File not found."
22
23 # 3 Invalid Path
24 assert read_file("invalid:/path/test.txt") == "Error: Invalid file path or unable to access file."
25
26 print("All test cases passed successfully!")
27
```

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE PORTS POSTMAN CONSOLE

Python Debug Console + ×

```
PS D:\python_dsa> & 'c:\Python314\python.exe' 'c:\users\yarav\vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '49701' '--' 'd:\python_dsa\AIAC.py'
y'
All test cases passed successfully!
PS D:\python_dsa>
```

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.

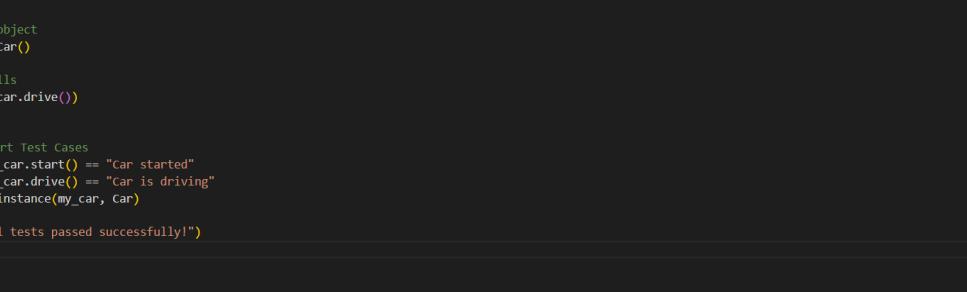
```
AIAC.py > ...
1 # Bug: Calling an undefined method
2 class Car:
3     def start(self):
4         return "Car started"
5
6 my_car = Car()
7 print(my_car.drive()) # drive() is not defined
8
```

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE PORTS POSTMAN CONSOLE

Python Debug Console + ×

```
y'
Traceback (most recent call last):
File "d:\python_dsa\AIAC.py", line 7, in <module>
print(my_car.drive()) # drive() is not defined
~~~~~
AttributeError: 'Car' object has no attribute 'drive'
PS D:\python_dsa>
```

Corrected Code



The screenshot shows a Python file named AIAC.py open in VS Code. The code defines a Car class with methods start() and drive(). It includes test cases for these methods and prints a success message if all tests pass.

```
AIAC.py
6     |     return "Car is driving"
7
8
9 # Create object
10 my_car = Car()
11
12 # Test calls
13 print(my_car.drive())
14
15
16 # Assert Test Cases
17 assert my_car.start() == "Car started"
18 assert my_car.drive() == "Car is driving"
19 assert isinstance(my_car, Car)
20
21 print("All tests passed successfully!")
22
23
24
25
26
27
```

VS Code interface elements include tabs for PROBLEMS, TERMINAL, OUTPUT, DEBUG CONSOLE, PORTS, and POSTMAN CONSOLE at the bottom left. At the bottom right, there's a Python Debug Console tab and a set of standard terminal navigation icons.

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

The screenshot shows a VS Code interface with the following details:

- Title Bar:** AIAC.py
- Code Editor:** The file contains Python code with a bug. Line 5 has a syntax error: `print(add_five("10"))` instead of `print(addFive("10"))`.
- Terminal:** The terminal window shows the command `PS D:\python_dsa>` followed by the stack trace and error message.
- Bottom Navigation:** PROBLEMS, TERMINAL, OUTPUT, DEBUG CONSOLE, PORTS, POSTMAN CONSOLE.
- Right Side:** A Python Debug Console tab is visible.

```
PS D:\python_dsa> ^C
PS D:\python_dsa>
PS D:\python_dsa> d; cd "d:\python_dsa"; & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' 'd:\python_dsa\AIAC.py'
Traceback (most recent call last):
  File "d:\python_dsa\AIAC.py", line 5, in <module>
    print(add_Five("10"))
           ^~~~~~^~~~~~
  File "d:\python_dsa\AIAC.py", line 3, in add_Five
    return value + 5
           ^~~~~~
TypeError: can only concatenate str (not "int") to str
AIAC.py
```

Solution 1: Type Casting (Convert to Integer)

A screenshot of the Visual Studio Code interface. The top half shows a code editor with the file 'AIAC.py' open. The code contains a function 'add_five' that adds 5 to an integer input. It includes test cases for '10', '20', and '0'. The bottom half shows a terminal window with the output of running the script, which prints 'Solution 1 tests passed!'.

```
AIAC.py > ...
1 # Solution 1: Type Casting
2 def add_five(value):
3     return int(value) + 5
4
5
6 # Test Cases
7 assert add_five("10") == 15
8 assert add_five(20) == 25
9 assert add_five("0") == 5
10 print("Solution 1 tests passed!")
11
12
13
14
15
16
17
18
19
```

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE PORTS POSTMAN CONSOLE Python Debug Console

```
PS D:\python_dsa> & 'c:\Python314\python.exe' 'c:\Users\yarav\vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '59738' '--' 'd:\pyt
'y'
Solution 1 tests passed!
PS D:\python_dsa>
```

Solution 2: String Concatenation

A screenshot of the Visual Studio Code interface. The top half shows a code editor with the file 'AIAC.py' open. The code contains a function 'add_five' that concatenates '5' to the end of a string input. It includes test cases for '10', '20', and an empty string. The bottom half shows a terminal window with the output of running the script, which prints 'solution 2 tests passed!'.

```
AIAC.py > ...
1 # Solution 2: String Concatenation
2 def add_five(value):
3     return str(value) + "5"
4
5
6 # Test Cases
7 assert add_five("10") == "105"
8 assert add_five(20) == "205"
9 assert add_five("") == "5"
10 print("solution 2 tests passed!")
11
12
13
14
15
16
17
18
19
20
```

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE PORTS POSTMAN CONSOLE Python Debug Console

```
PS D:\python_dsa> & 'c:\Python314\python.exe' 'c:\Users\yarav\vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '63944' '--' 'd:\pyt
'y'
Solution 2 tests passed!
PS D:\python_dsa>
```