AI ASSISTED CODING

ASSIGNMENT 7

2403A51236-SIRI

BATCH 11

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.

Code with error:

Prompt: Explain the error and fix it.

AI explanation:

The error is caused by using Python 2 style print syntax (print "Hello, AI Debugging Lab!").
In Python 3, print is a function and requires parentheses.

Corrected code:

Task Description #2 (Logic Error – 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.

Code with error:

```
task2.py > check_number
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"
7
```

```
PROBLEMS 2    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                    Python + ...  []  X

PS C:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8> & C:/Users/siris/AppData/Local/Microsoft
/WindowsApps/python3.11.exe c:/Users/siris/OneDrive/Documents/AI-Assignments/assignment7,8/task2.py
  File "c:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8\task2.py", line 3
    if n = 10:
        ^^^^^^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
PS C:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8>
```

Prompt: Explain the error and fix it.

Error Explanation:
The code uses the assignment operator = instead of the comparison operator == in the if statement.
In Python, = is used to assign values, while == checks for equality.
Using = in a conditional statement causes a SyntaxError.

CORRECTED CODE:

```
task2.py > check_number
1    def check_number(n):
2        if n == 10:  # ✅ Use comparison operator
3            return "Ten"
4        else:
5            return "Not Ten"
```

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

Task: Provide code that attempts to open a non-existent file and crashes.

```
TASK3.PY > read_file
1    # Bug: Program crashes if file is missing
2    def read_file(filename):
3        with open(filename, 'r') as f:
4            return f.read()
5
6    print(read_file("nonexistent.txt"))
7
```

```
PROBLEMS 1    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                          Python + v  ...  [] ×

PS C:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8> & C:/Users/siris/AppData/Local/Microsoft
/WindowsApps/python3.11.exe c:/Users/siris/OneDrive/Documents/AI-Assignments/assignment7,8/TASK3.PY
Traceback (most recent call last):
  File "c:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8\TASK3.PY", line 6, in <module>
    print(read_file("nonexistent.txt"))
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8\TASK3.PY", line 3, in read_file
    with open(filename, 'r') as f:
         ^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'nonexistent.txt'
PS C:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8>
```

Prompt: Explain the error and fix it.

Error Explanation:
If "nonexistent.txt" does not exist, open(filename, 'r') raises a FileNotFoundError, causing the program to crash.

Fix:
Wrap the file opening code in a try...except block to handle missing files gracefully.

Corrected code:

```
TASK3.PY > read_file
1    # Bug: Program crashes if file is missing
2    def read_file(filename):
3        try:
4            with open(filename, 'r') as f:
5                return f.read()
6        except FileNotFoundError:
7            return "File not found."
8
9    print(read_file("nonexistent.txt"))
10
```

```
PROBLEMS 1    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                          Python + v  ...  [] ×

PS C:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8> & C:/Users/siris/AppData/Local/Microsoft
/WindowsApps/python3.11.exe c:/Users/siris/OneDrive/Documents/AI-Assignments/assignment7,8/TASK3.PY
File not found.
PS C:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8>
```

Task Description #4 (AttributeError – 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.

```
task4.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())
8
```

```
PROBLEMS 1    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8> & C:/Users/siris/AppData/Local/Microsoft
/WindowsApps/python3.11.exe c:/Users/siris/OneDrive/Documents/AI-Assignments/assignment7,8/task4.py
Traceback (most recent call last):
  File "c:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8\task4.py", line 7, in <module>
    print(my_car.drive())
          ^^^^^^^^^^^^
AttributeError: 'Car' object has no attribute 'drive'
PS C:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8>
```

Prompt: Explain the error and fix it

Error Explanation:
The code tries to call my_car.drive(), but the Car class does not have
a drive method defined.
This causes an AttributeError.

Corrected code:

```
task4.py > ...
1    # Bug: Calling an undefined method
2    class Car:
3        def start(self):
4            return "Car started"
5
6        def drive(self):
7            return "Car is driving"
8
9    my_car = Car()
10   print(my_car.drive())
11
```

```
PROBLEMS 1    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8> & C:/Users/siris/AppData/Local/Microsoft
/WindowsApps/python3.11.exe c:/Users/siris/OneDrive/Documents/AI-Assignments/assignment7,8/task4.py
Car is driving
PS C:\Users\siris\OneDrive\Documents\AI-Assignments\assignment7,8>
```

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.

Prompt: Explain the error and fix it

Error Explanation:
The code tries to add an integer (5) to a string ("10"), which causes
a TypeError in Python.

Fix:
Convert the input to an integer before adding.

Corrected code: