SR UNIVERSITY

AI ASSIST CODING

Lab-7.4:

ROLL NO:2503A51L17

NAME: Simra Tahseen

Batch: 24B2CAICSB19

Lab Objectives:

- 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.

Lab Outcomes (LOs):

After completing this lab, students will be able to:

- 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.

Task 1:

Prompt Used:

Fix this recursive factorial function by detecting the logical error and providing a correct implementation**.**

Buggy Code:

```
def factorial(n):

   return n * factorial(n-1)
```

Code entered:

```
mean.py > buggy.py > factorial
def factorial(n):
    return n * factorial(n-1)   # No base case
print(factorial(n))
```

```
PROBLEMS 2    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "
ocal/Programs/Python/Python37/python.exe" "c:/Users/SANIYA T
NG/mean.py/buggy.py"
Traceback (most recent call last):
  File "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING
odule>
    print(factorial(n))
NameError: name 'n' is not defined
```

Expected fix:

```
mean.py > buggy.py > ...
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

print(factorial(5))
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Python
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/Users/S
ocal/Programs/Python/Python37/python.exe" "c:/Users/SANIYA TAHSEEN/One
NG/mean.py/buggy.py"
120
```

Observation:

- Missing base case caused infinite recursion.
- AI detected the logical error.
- Fix added base case (n == 0 or n == 1).
- Corrected function now works for all positive integers.

Task 2:

Prompt Used:

Detect the type inconsistency in this sorting function and fix the code so it sorts the list consistently.

Code:

data = [5, "3", 7, "1"] print(sorted(data))

```
hello.py   for.py   stud.py   shop.py 1
mean.py > buggy.py > ...
data = [5, "3", 7, "1"]
print(sorted(data))  #  TypeError
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Pytho
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/User
ocal/Programs/Python/Python37/python.exe" "c:/Users/SANIYA TAHSEEN/
NG/mean
Traceba   Open file in editor (ctrl + click)   :
  File "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/mean.p
odule>
    print(factorial(n))
NameError: name 'n' is not defined
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/User
ocal/Programs/Python/Python37/python.exe" "c:/Users/SANIYA TAHSEEN/
NG/mean.py/buggy.py"
Traceback (most recent call last):
  File "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/mean.p
odule>
    print(sorted(data))  #  TypeError
TypeError: '<' not supported between instances of 'str' and 'int'
```

Expected Fix:

```
nean.py > buggy.py > ...
data = [5, "3", 7, "1"]
data = [int(x) for x in data]
print(sorted(data))
```

```
PROBLEMS 1    OUTPUT    DEBUG CONSOLE    TERMINAL    POR

PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_C
ocal/Programs/Python/Python37/python.exe" "c:/User
NG/mean.py/buggy.py"
[1, 3, 5, 7]
```

Observation:

- Mixed data types (int + str) caused a TypeError during sorting.
- AI identified the inconsistency.
- Fix applied by converting all elements to a common type (e.g., int).
- Corrected code sorts the list without errors.

Task 3:

Prompt Used:

Refactor this file handling code to use the best practice with a context manager and prevent resource leakage.

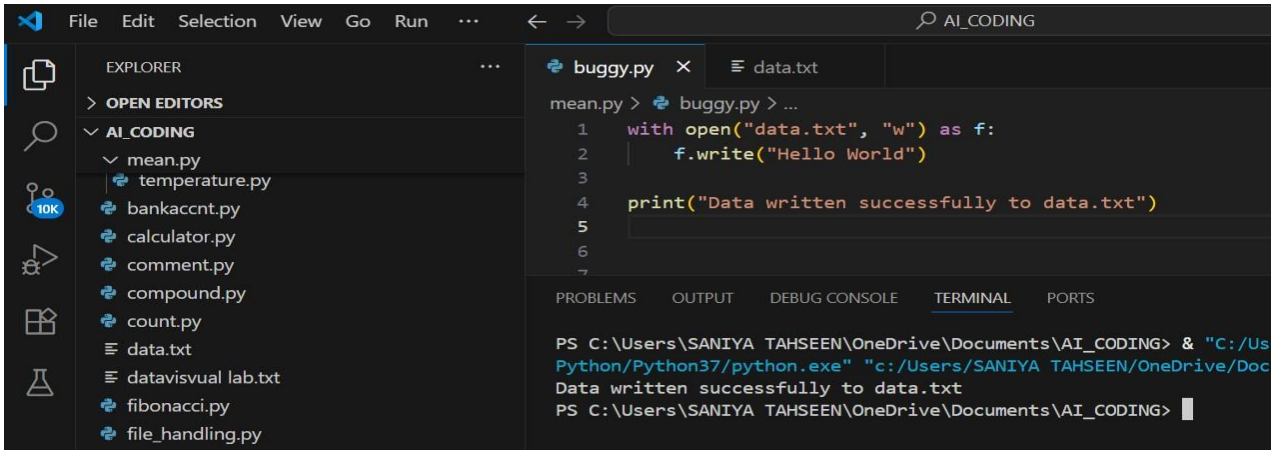
Code: f = open("data.txt",

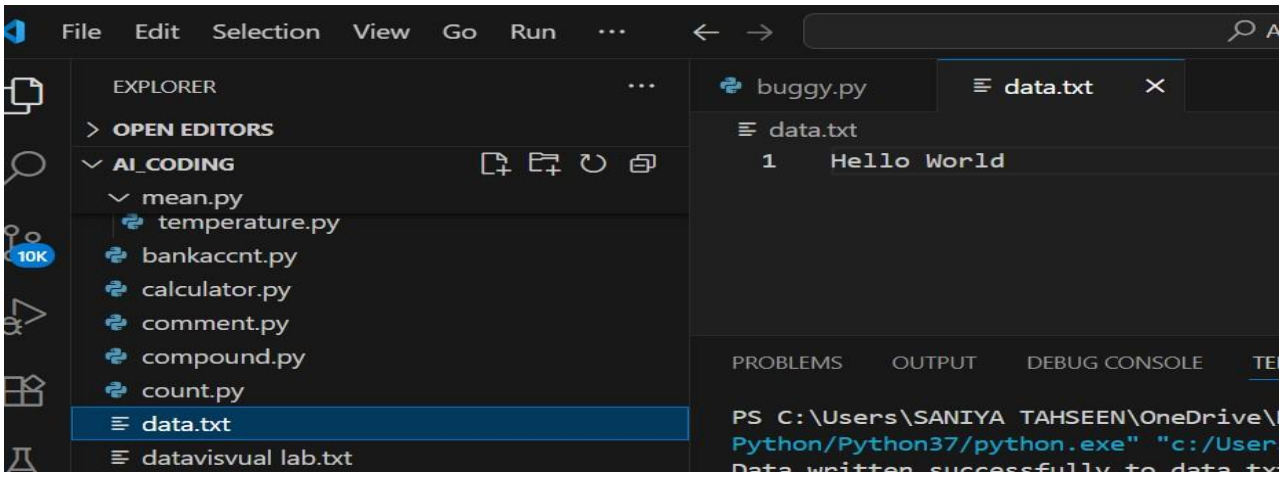
"w")

f.write("Hello World")

```
mean.py > buggy.py > ...
f = open("data.txt", "w")
f.write("Hello World")
```

Expected Fix:





Observation:

- data.txt created in project folder.
- Old content (if any) overwritten.
- "Hello World" written successfully.
- Confirmation message shown

Task 4:

Prompt Used:

Add proper error handling to this loop to prevent ZeroDivisionError by using try-except and display a safe error message.

Code:

```
for i in range(-2, 3):
    print(10 / i)  #  ZeroDivisionError
```

Code Entered:

```
mean.py > error.py > ...
for i in range(-2, 3):
    print(10 / i)  # ZeroDivisionError
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/Users/SANIYA TAHSEEN/AppData/Local/Pro
/python.exe" "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/mean.py/error.py"
-5.0
-10.0
Traceback (most recent call last):
  File "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/mean.py/error.py", line 2, in <module>
    print(10 / i)  #  ZeroDivisionError
ZeroDivisionError: division by zero
```

Expected Fix:

```
mean.py > error.py > ...
for i in range(-2, 3):
    try:
        print(10 / i)
    except ZeroDivisionError:
        print("Division by zero is not allowed")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\AI_CODING> & "C:/Users/SAN
/python.exe" "c:/Users/SANIYA TAHSEEN/OneDrive/Documents/AI_CODING/mean.
-5.0
-10.0
Division by zero is not allowed
10.0
5.0
```

Observation:

- Division by zero occurred inside the loop.
- AI detected the runtime error.
- Added try-except block to handle ZeroDivisionError.
- Now prints a safe message instead of crashing.

Task 5:

Prompt Used:

Correct the constructor and attribute references in this class definition by fixing mismatched parameters and ensuring proper initialization.

Code:

class Student:

def __init__(self, name, agee):

self.name = nme self.age =

age # mismatch **code entered:**

```
buggy.py        error.py 2 ×        data.txt
ean.py > error.py > Student > __init__
class Student:
    def __init__(self, name, agee):
        self.name = nme    # typo
        self.age = age     #  mismatch
```

Expected Fix:

```
buggy.py        error.py ×        data.txt
mean.py > error.py > ...
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age

s1 = Student("Alice", 20)
print(s1.name, s1.age)

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    POR

PS C:\Users\SANIYA TAHSEEN\OneDrive\Documents\A
/python.exe" "c:/Users/SANIYA TAHSEEN/OneDrive/
Alice 20
```

Observation:

- Constructor had typos (nme, agee) and mismatched variables.
- AI detected and corrected parameter/attribute issues.
- Fixed class initializes attributes properly.
- Objects now create successfully and display expected values.