AI ASSISTED CODING

ASSIGNMENT-10.1

Name: Harshitha Guda

H.T.No:2303A51102

Task Description #1 – Syntax and Logic Errors

Task: Use AI to identify and fix syntax and logic errors in a faulty Python script.

Sample Input Code:

Calculate average score of a student

def calc_average(marks):

total = 0

for m in marks:

total += m

average = total / len(marks)

return avrage # Typo here

marks = [85, 90, 78, 92]

print("Average Score is ", calc_average(marks)

Expected Output:

• Corrected and runnable Python code with explanations of the fixes.

```
1   # Calculate average score of a student
2   def calc_average(marks):
3       total = 0
4       for m in marks:
5           total += m
6       average = total / len(marks)
7       return average # Typo here
8   marks = [85, 90, 78, 92]
9   print("Average Score is ", calc_average(marks))
10
11
12
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\gudah\OneDrive\Documents\AIAC> & C:\Python314\python.exe c:/Users/gudah/OneDrive/Documents/AIAC/Lab_assignment_10.1.py
Average Score is  86.25
PS C:\Users\gudah\OneDrive\Documents\AIAC>
```

Task Description #2 – PEP 8 Compliance

Task: Use AI to refactor Python code to follow PEP 8 style guidelines.

Sample Input Code:

def area_of_rect(L,B) : return L*B

print(area_of_rect(10,20))

Expected Output:

• Well-formatted PEP 8-compliant Python code

```
13  def area_of_rect(length: float, width: float) -> float:
14      """Calculate the area of a rectangle.
15      Args:
16      length (float): The length of the rectangle.
17      width (float): The width of the rectangle."""
18      return length * width
19  length = 5.0
20  width = 3.0
21  print("Area of the rectangle is ", area_of_rect(length, width))
```

Task Description #3 – Readability Enhancement

Task: Use AI to make code more readable without changing its

logic.

Sample Input Code:

def c(x,y):

return x*y/100

a=200

b=15

print(c(a,b))

Expected Output:

• Python code with descriptive variable names, inline

comments, and clear formatting.

```
25  def calculate_percentage(total: float, percentage: float) -> float:
26      """Calculate the percentage of a total value."""
27      return total * percentage / 100# Example usage
28
29  # Calculate 15% of 200
30  total_amount = 200# Percentage rate to calculate
31  percentage_rate = 15# Call the function and print the result
32  result = calculate_percentage(total_amount, percentage_rate)# Print the result
33  print(f"Result: {result}")# Output: Result: 30.0
34
```

Task Description #4 – Refactoring for Maintainability

Task: Use AI to break repetitive or long code into reusable

functions.

Sample Input Code:

students = ["Alice", "Bob", "Charlie"]

print("Welcome", students[0])

print("Welcome", students[1])

print("Welcome", students[2])

Expected Output:

• Modular code with reusable functions.

```
def welcome_students(students):
    """Welcome each student in the list."""
    for student in students:
        print(f"Welcome, {student}")

students = ["Alice", "Bob", "Charlie"]
welcome_students(students)

```

Task Description #5 – Performance Optimization

Task: Use AI to make the code run faster.

Sample Input Code:

Find squares of numbers

nums = [i for i in range(1,1000000)]

squares = []

for n in nums:

squares.append(n**2)

print(len(squares))

Expected Output:

• Optimized code using list comprehensions or vectorized

operations.

```
# Create a list of integers from 1 to 999,999
nums = [i for i in range(1, 1000000)]

# Create a list of squares for each number in nums
squares = [n**2 for n in nums]

# Print the length of the squares list
print(len(squares))

```

Task Description #6 – Complexity Reduction

Task: Use AI to simplify overly complex logic.

Sample Input Code:

def grade(score):

if score >= 90:

return "A"

else:

if score >= 80:

return "B"

else:

if score >= 70:

return "C"

else:

if score >= 60:

return "D"

else:

return "F"

Expected Output:

• Cleaner logic using elif or dictionary mapping.

```
def grade(score):
    if score >= 90:
        return "A"
    elif score >= 80:
        return "B"
    elif score >= 70:
        return "C"
    elif score >= 60:
        return "D"
    else:
        return "F"
# Example usage
student_score = 85
print(f"The grade for a score of {student_score} is: {grade(student_score)}")

```