

AI-ASSISTED CODING

ASSIGNMENT-3

Task Description #1 (AI-Based Code Completion for Loops)

Task: Use an AI code completion tool to generate a loop-based program.

Prompt:

"Generate Python code to print all even numbers between 1 and N using a loop."

Expected Output:

- AI-generated loop logic.
- Identification of loop type used (for or while).
- Validation with sample inputs.

Prompt:

```
#Generate Python code to print all even numbers between 1 and N using a loop
```

Code:

```
def print_even_numbers(N):  
    for num in range(1, N + 1):  
        if num % 2 == 0:  
            print(num)  
N = int(input("Enter a number N: "))  
print_even_numbers(N)
```

Output:

```
PS C:\Users\nagas\OneDrive\Desktop\AI Lab> & C:/Users/nagas/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/nagas/OneDrive/Desktop/AI Lab/Lab4.py"  
Enter a number N: 8  
2  
4  
6  
8  
PS C:\Users\nagas\OneDrive\Desktop\AI Lab> █
```

Analysis:

AI generated a loop using a **for** loop to correctly print even numbers between 1 and N.

Task Description #2 (AI-Based Code Completion for Loop with Conditionals)

Task: Use an AI code completion tool to combine loops and conditionals.

Prompt:

"Generate Python code to count how many numbers in a list are even and odd."

Expected Output:

- AI-generated code using loop and if condition.
- Correct count validation.
- Explanation of logic flow.

Prompt:

```
#Generate Python code to count how many numbers in a list are even and odd
```

Code:

```
def count_even_odd(numbers):  
    even_count = 0  
    odd_count = 0  
    for num in numbers:  
        if num % 2 == 0:  
            even_count += 1  
        else:  
            odd_count += 1  
    return even_count, odd_count  
  
numbers = [int(x) for x in input("Enter numbers separated by spaces: ").split()]  
even_count, odd_count = count_even_odd(numbers)  
print(f"Even numbers count: {even_count}")  
print(f"Odd numbers count: {odd_count}")
```

Output:

```
PS C:\Users\nagas\OneDrive\Desktop\AI Lab> & C:/Users/nagas/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/nagas/OneDrive/Desktop/AI Lab/Lab4.py"  
Enter numbers separated by spaces: 1 2 3 4 5 34 27 5 7 8 9  
Even numbers count: 4  
Odd numbers count: 7  
PS C:\Users\nagas\OneDrive\Desktop\AI Lab>
```

Analysis:

AI used a loop with `if-else` conditions to accurately count even and odd numbers in a list.

Task Description #3 (AI-Based Code Completion for Class Attributes Validation)

Task: Use an AI tool to complete a Python class that validates user input.

Prompt:

"Generate a Python class User that validates age and email using conditional statements."

Expected Output:

- AI-generated class with validation logic.
- Verification of condition handling.
- Test cases for valid and invalid inputs.

Prompt:

```
#Generate a Python class User that validates age and email using conditional statements
```

Code:

```
import re
class User:
    def __init__(self, name, age, email):
        self.name = name
        self.age = age
        self.email = email
        self.validate_age()
        self.validate_email()

    def validate_age(self):
        if not (0 < self.age < 120):
            raise ValueError("Age must be between 1 and 119.")

    def validate_email(self):
        email_pattern =
r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
        if not re.match(email_pattern, self.email):
            raise ValueError("Invalid email format.")

User_name = input("Enter your name: ")
User_age = int(input("Enter your age: "))
User_email = input("Enter your email: ")
try:
    user = User(User_name, User_age, User_email)
    print("User created successfully.")
except ValueError as e:
    print(f"Error: {e}")
```

Output:

```
PS C:\Users\nagas\OneDrive\Desktop\AI Lab> & C:/Users/nagas/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/nagas/OneDrive/Desktop/AI Lab/Lab4.py"
Enter your name: Ngasri
Enter your age: 20
Enter your email: 2303A51247@sru.edu.in
User created successfully.
PS C:\Users\nagas\OneDrive\Desktop\AI Lab> |
```

Analysis:

AI created a `User` class that validates age and email using conditional checks.

Task Description #4 (AI-Based Code Completion for Classes)

Task: Use an AI code completion tool to generate a Python class for managing student details.

Prompt:

“Generate a Python class Student with attributes (name, roll number, marks) and methods to calculate total and average marks.”

Expected Output:

- AI-generated class code.
- Verification of correctness and completeness of class structure.
- Minor manual improvements (if needed) with justification.

Prompt:

```
#Generate a Python class Student with attributes (name, roll number,marks) and methods to calculate total and average marks.
```

Code:

```
class Student:
    def __init__(self, name, roll_number, marks):
        self.name = name
        self.roll_number = roll_number
        self.marks = marks

    def calculate_total(self):
        return sum(self.marks)

    def calculate_average(self):
        if len(self.marks) == 0:
            return 0
        return self.calculate_total() / len(self.marks)
```

```

student_name = input("Enter student name: ")
student_roll_number = input("Enter roll number: ")
marks_input = input("Enter marks separated by spaces: ")
marks = [float(mark) for mark in marks_input.split()]
student = Student(student_name, student_roll_number, marks)
total_marks = student.calculate_total()
average_marks = student.calculate_average()
print(f"Total Marks: {total_marks}")
print(f"Average Marks: {average_marks}")

```

Output:

```

PS C:\Users\nagas\OneDrive\Desktop\AI Lab> & C:/Users/nagas/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/nagas/OneDrive/Desktop/AI Lab/ab4.py"
Enter student name: nagasri
Enter roll number: 1247
Enter marks separated by spaces: 75 85 95 90 50
Total Marks: 395.0
Average Marks: 79.0
PS C:\Users\nagas\OneDrive\Desktop\AI Lab>

```

Analysis:

AI generated a `Student` class with methods to calculate total and average marks correctly.

Task Description 5 (AI-Assisted Code Completion Review)

Task: Use an AI tool to generate a complete Python program using classes, loops, and conditionals together.

Prompt:

"Generate a Python program for a simple bank account system using class, loops, and conditional statements."

Prompt:

```

#Generate a Python program for a simple bank account system using
class, loops, and conditional statements

```

Code:

```

class BankAccount:
    def __init__(self, account_holder, balance=0):
        self.account_holder = account_holder

```

```

        self.balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited: ${amount:.2f}")
        else:
            print("Deposit amount must be positive.")

    def withdraw(self, amount):
        if 0 < amount <= self.balance:
            self.balance -= amount
            print(f"Withdrew: ${amount:.2f}")
        else:
            print("Insufficient funds or invalid withdrawal amount.")

    def display_balance(self):
        print(f"Account Holder: {self.account_holder}, Balance: ${self.balance:.2f}")

account_holder = input("Enter account holder name: ")
account = BankAccount(account_holder)

while True:
    print("\nOptions: 1. Deposit 2. Withdraw 3. Display Balance 4. Exit")

    choice = input("Choose an option (1-4): ")
    if choice == '1':
        amount = float(input("Enter amount to deposit: "))
        account.deposit(amount)
    elif choice == '2':
        amount = float(input("Enter amount to withdraw: "))
        account.withdraw(amount)
    elif choice == '3':
        account.display_balance()
    elif choice == '4':
        print("Exiting the program.")
        break
    else:
        print("Invalid choice. Please try again.")

```

Output:

```
PS C:\Users\nagas\OneDrive\Desktop\AI Lab> & C:/Users/nagas/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/nagas/OneDrive/Desktop/AI Lab/Lab4.py"
Enter account holder name: nagasri

Options: 1. Deposit 2. Withdraw 3. Display Balance 4. Exit
Choose an option (1-4): 1
Enter amount to deposit: 5000
Deposited: $5000.00

Options: 1. Deposit 2. Withdraw 3. Display Balance 4. Exit
Choose an option (1-4): 2
Enter amount to withdraw: 579
Withdrew: $579.00

Options: 1. Deposit 2. Withdraw 3. Display Balance 4. Exit
Choose an option (1-4): 3
Account Holder: nagasri, Balance: $4421.00

Options: 1. Deposit 2. Withdraw 3. Display Balance 4. Exit
Choose an option (1-4): 4
Exiting the program.
PS C:\Users\nagas\OneDrive\Desktop\AI Lab> █
```

Analysis:

AI helped generate a complete bank account program using classes, loops, and conditionals, improving coding speed and structure.