

## AI ASSISTED CODING

### ASSIGNMENT 6.1

Name: Ch. Tejaswi

Roll no: 2303A51944

Batch :27

#### Task Description #1: AI-Based Code Completion for Loops

##### Prompt Used

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

```
python

n = int(input("Enter a number: "))

for i in range(1, n + 1):
    if i % 2 == 0:
        print(i)
```

**Input:**

```
10
```

**Output:**

```
2
4
6
8
10
```

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

##### Prompt Used

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

## AI-Generated Code

```
python

numbers = [1, 2, 3, 4, 5, 6]
even_count = 0
odd_count = 0

for num in numbers:
    if num % 2 == 0:
        even_count += 1
    else:
        odd_count += 1

print("Even numbers:", even_count)
print("Odd numbers:", odd_count)
```

## Output:

```
yaml
```

```
Even numbers: 3
Odd numbers: 3
```

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

### Prompt Used

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

## AI-Generated Code

```
python

class User:
    def __init__(self, age, email):
        self.age = age
        self.email = email

    def validate(self):
        if self.age < 18:
            print("Invalid age")
        elif "@" not in self.email:
            print("Invalid email")
        else:
            print("User details are valid")
```

## Test Cases

### Valid Input

```
python

u1 = User(22, "test@gmail.com")
u1.validate()
```

**Output:** User details are valid

### Task Description #4: AI-Based Code Completion for Classes

#### Prompt Used

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

## AI-Generated Code

```
python

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

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

    def average_marks(self):
        return self.total_marks() / len(self.marks)
```

### Task Description #5: AI-Assisted Code Completion Review

#### Prompt Used

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

## AI-Generated Program

```
python

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

    def deposit(self, amount):
        self.balance += amount

    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
        else:
            print("Insufficient balance")

    def show_balance(self):
        print("Balance:", self.balance)
```

```
account = BankAccount()

while True:
    print("1.Deposit 2.Withdraw 3.Balance 4.Exit")
    choice = int(input("Enter choice: "))

    if choice == 1:
        amount = int(input("Enter amount: "))
        account.deposit(amount)
    elif choice == 2:
        amount = int(input("Enter amount: "))
        account.withdraw(amount)
    elif choice == 3:
        account.show_balance()
    elif choice == 4:
        break
    else:
        print("Invalid choice")
```