School of Computer Science and Artificial Intelligence

Lab Assignment # 6.5

Program : B. Tech (CSE)

Specialization :AIML

Course Title : AI Assisted Coding

Course Code : 23CS002PC304

Semester : VI

Academic Session : 2025-2026

Name of Student : R.Sowmya Sri

Enrollment No. : 2303A52105

Batch No. : 33

Date :30/01/26

Experiment 6: AI-Based Code Completion

Aim

To use AI-based code completion tools to generate, analyze, optimize, and ethically evaluate Python programs involving classes, loops, and conditional statements.

Learning Outcomes Addressed

- LO1: Generate Python code using AI
- LO2: Explain AI-generated code line-by-line
- LO3: Identify logical flaws or inefficiencies
- LO4: Optimize AI-generated code
- LO5: Demonstrate ethical use of AI tools

Task 1: Conditional Eligibility Check (Voting Eligibility)

Prompt:

"Generate Python code to check voting eligibility based on age and citizenship."

Code:

```
age = int(input("Enter your age: "))
citizen = input("Are you a citizen? (yes/no): ").lower()

if age >= 18 and citizen == "yes":
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote.")
```

Output:

```
Enter your age: 20
Are you a citizen? (yes/no): yes
You are eligible to vote.
```

Explanation:

- Takes age as integer input
- Takes citizenship status as string
- Uses **AND condition**:
  - Age must be **18 or above**
  - Citizenship must be **yes**
- Prints eligibility result

Task 2: Loop-Based String Processing (Vowels & Consonants)

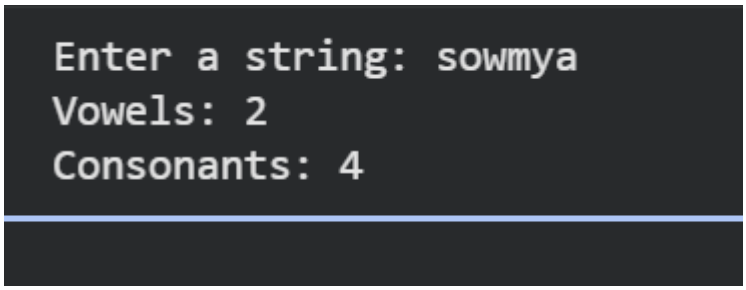
"Generate Python code to count vowels and consonants in a string using a loop."

Code:

```
text = input("Enter a string: ").lower()
vowels = "aeiou"
vowel_count = 0
consonant_count = 0

for char in text:
    if char.isalpha():
        if char in vowels:
```

```
        vowel_count += 1
    else:
        consonant_count += 1

print("Vowels:", vowel_count)
print("Consonants:", consonant_count)
```

Output:



Explanation:

- Converts string to lowercase
- Loops through each character
- Checks:
  - Alphabet or not
  - Vowel or consonant
- Counts correctly

Task 3: AI-Generated Library Management System

Prompt:

"Generate a Python program for a library management system using classes, loops, and conditional statements."

Code:

```
class Library:
```

```
    def __init__(self):
        self.books = []

    def add_book(self, book):
        self.books.append(book)
        print(f"'{book}' added to library.")

    def display_books(self):
        if not self.books:
            print("No books available.")
        else:
            print("Books in library:")
            for book in self.books:
                print("-", book)


library = Library()

while True:
    print("\n1. Add Book")
    print("2. Display Books")
    print("3. Exit")

    choice = input("Enter choice: ").lower().strip()

    if choice in ["1", "add book", "add"]:
```

```
        book = input("Enter book name: ")
        library.add_book(book)

    elif choice in ["2", "display books", "display"]:
        library.display_books()

    elif choice in ["3", "exit"]:
        print("Exiting program.")
        break

    else:
        print("Invalid choice. Please try again.")
```

Output:

```
1. Add Book
2. Display Books
3. Exit
Enter choice: 1
Enter book name: Harry Potter
'Harry Potter' added to library.

1. Add Book
2. Display Books
3. Exit
Enter choice: 1
Enter book name: Hunger Games
'Hunger Games' added to library.

1. Add Book
2. Display Books
3. Exit
Enter choice: 2
Books in library:
- Harry Potter
- Hunger Games

1. Add Book
2. Display Books
3. Exit
```

```
1. Add Book
2. Display Books
3. Exit
Enter choice: 3
Exiting program.
```

Explaination:

AI greatly speeds up initial code creation and helps structure programs logically. However, human review is essential to verify correctness, improve features, and ensure ethical use without blindly trusting generated code.

Task 4: Class-Based Attendance System

Prompt:

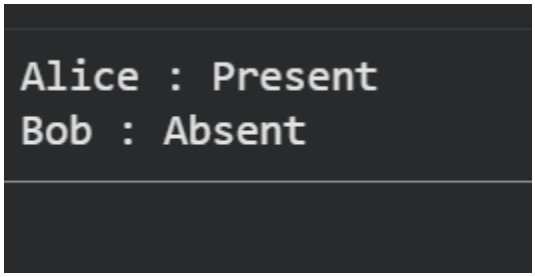
"Generate a Python class to mark and display student attendance using loops."

Code:

```
class Attendance:
    def __init__(self):
        self.students = {}

    def mark_attendance(self, name, status):
        self.students[name] = status

    def display_attendance(self):
        for name, status in self.students.items():
```

```
        print(name, ":", status)


attendance = Attendance()


attendance.mark_attendance("Alice", "Present")
attendance.mark_attendance("Bob", "Absent")
attendance.display_attendance()
```

Output:



Explanation:

-Dictionary stores student attendance

- Loop prints all records

- Class improves data organization

Task 5: ATM Menu Simulation (Loops & Conditionals)

Prompt:

"Generate a Python program using loops and conditionals to simulate an ATM menu."

Code:

```
balance = 5000


while True:
```

```
print("\nATM Menu")
print("1. Check Balance")
print("2. Deposit")
print("3. Withdraw")
print("4. Exit")

choice = input("Enter choice: ")

if choice == "1":
    print("Balance:", balance)

elif choice == "2":
    amount = int(input("Enter deposit amount: "))
    balance += amount
    print("Amount deposited.")

elif choice == "3":
    amount = int(input("Enter withdrawal amount: "))
    if amount <= balance:
        balance -= amount
        print("Withdrawal successful.")
    else:
        print("Insufficient balance.")

elif choice == "4":
    print("Thank you!")
```

```
        break

    else:
        print("Invalid option.")
```

Output:

```
ATM Menu
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter choice: 2
Enter deposit amount: 20000
Amount deposited.

ATM Menu
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter choice: 1
Balance: 25000

ATM Menu
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter choice: 3
Enter withdrawal amount: 10000
Withdrawal successful.
```

Explanation:

- AI was used as a **coding assistant**, not a replacement
- Code was **reviewed, tested, and optimized manually**
- No blind copying—logic was understood and explained
- Encourages learning, not dependency