AI ASSISTANT CODING

ASSIGNMENT– 6

NAME:SUMANTH AKARAPU

H.NO:2303A52191

Batch-34

Title

AI-Based Code Completion: Working with Suggestions for Classes, Loops, and Conditionals

---

Aim

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

Learning Objectives

- **LO1:** Use AI-based tools to generate Python code using classes, loops, and conditionals
- **LO2:** Interpret and explain AI-generated code line by line
- **LO3:** Identify errors and inefficiencies in AI-generated code
- **LO4:** Optimize AI-generated code for readability and performance
- **LO5:** Demonstrate ethical and responsible use of AI tools

---

Tools Used

- Python 3.x
- AI-based code completion tool (ChatGPT)

Task 1: AI-Based Code Completion for Conditional Eligibility Check

Prompt Used

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

Program

```
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.")
```

Explanation

- Takes user input for age and citizenship
- Uses conditional statements to verify eligibility
- Displays appropriate output based on conditions

Output

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

Task 2: AI-Based Code Completion for Loop-Based String Processing

Prompt Used

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

Program

```
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)
```

Explanation

- Iterates through each character using a loop
- Counts vowels and consonants separately
- Ignores spaces and special characters

Output

```
Enter a string: Hello World
Vowels: 3
Consonants: 7
```

Task 3: AI-Assisted Code Completion Reflection Task

Prompt Used

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

Program

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

    def add_book(self, book):
        self.books.append(book)

    def display_books(self):
        if not self.books:
            print("No books available.")
        else:
            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: ")

    if choice == "1":
        book = input("Enter book name: ")
        library.add_book(book)
    elif choice == "2":
        library.display_books()
    elif choice == "3":
```

```
        break
    else:
        print("Invalid choice.")
```

Review of AI Suggestions

- Correct use of classes and objects
- Simple and readable logic
- Menu-driven program using loops

Reflection

AI-assisted coding helps in generating quick and structured solutions. However, reviewing and optimizing AI-generated code is necessary to ensure correctness, efficiency, and ethical use.

Task 4: AI-Based Code Completion for Class-Based Attendance System

Prompt Used

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

Program

```
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

```
Alice : Present
Bob : Absent
```

Task 5: AI-Based Code Completion for Conditional Menu Navigation

Prompt Used

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

Program

```
balance = 1000

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
    elif choice == "3":
        amount = int(input("Enter withdrawal amount: "))
        if amount <= balance:
```

```
            balance -= amount
        else:
            print("Insufficient balance.")
    elif choice == "4":
        break
    else:
        print("Invalid choice.")
```

Ethical Use of AI

- AI was used as a support tool, not a replacement for understanding
- Generated code was reviewed and tested
- Logical errors were identified and corrected
- Responsible and ethical use of AI was followed

Conclusion

This experiment successfully demonstrated the use of AI-based code completion to generate Python programs involving conditionals, loops, and classes. AI tools improve coding efficiency, but human analysis is essential for validation and optimization.