Hepsiba Devara

2403A51L25

Batch-51

## Experiment 6: AI-Based Code Completion

**Topic:** Classes, Loops, Conditionals (Python)

### Task 1: Voting Eligibility Check (Conditionals)

**Prompt used in AI tool:**

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

### Python Code:

```
6_Ass.py > ...
1   age = int(input("Enter age: "))
2   citizen = input("Are you a citizen? (yes/no): ").lower
3
4   if age >= 18 and citizen == "yes":
5       print("Eligible to vote")
6   else:
7       print("Not eligible to vote")
8
```

### Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

ocal/Programs/Python/Python313/python.exe c:/Users/hepsi/OneDrive/Deskt
Enter age: 18
Are you a citizen? (yes/no): yes
Eligible to vote
PS C:\Users\hepsi\OneDrive\Desktop\AI>
```

### Explanation:

- age >= 18 → minimum voting age
- citizen == "yes" → must be a citizen
- and ensures **both conditions** are true

## Task 2: Count Vowels and Consonants (Loops)

**Prompt:**

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

## Python Code:

```
text = input("Enter a string: ").lower()
vowels = 0
consonants = 0

for ch in text:
    if ch.isalpha():
        if ch in "aeiou":
            vowels += 1
        else:
            consonants += 1

print("Vowels:", vowels)
print("Consonants:", consonants)
```

## Output:

```
PS C:\Users\hepsi\OneDrive\Desktop\AI> & C:/Users/hepsi/AppData/Local/Programs/Python/Python313/pyt
Enter a string: hepsiba
Vowels: 3
Consonants: 4
```

## Explanation:

for ch in text → loop through characters

isalpha () → ignores spaces/numbers

Counts vowels and consonants separately

## Task 3: Library Management System (Class + Loop + Conditions)

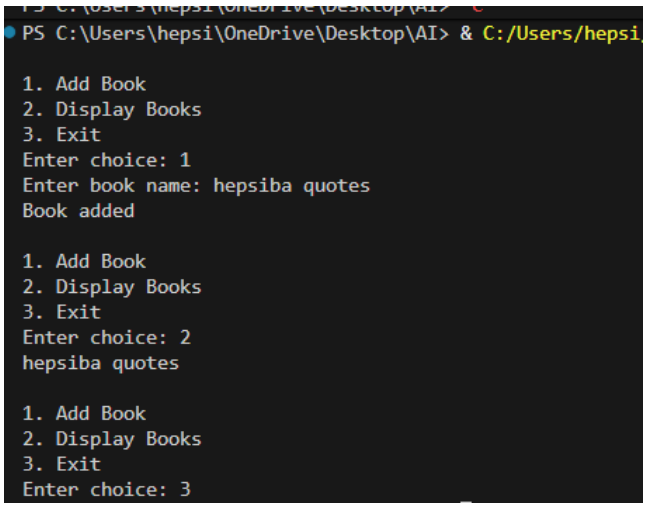
**Prompt:**

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

**Python code:**

```python
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 = int(input("Enter choice: "))

    if choice == 1:
        book = input("Enter book name: ")
        library.add_book(book)
        print("Book added")

    elif choice == 2:
        library.display_books()

    elif choice == 3:
        break

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

## Output:




## Explanation:

AI-based code completion helped generate a structured program quickly.
The logic was correct, but variable names and menu clarity were improved manually.
AI increases productivity but human review is essential.

## Task 4: Attendance Management System (Class + Loop)

**Prompt:**

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

## Python code:

```python
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:

```
PS C:\Users\hepsi\OneDrive\Desktop\AI> & C:/Users/hepsi/AppData/Loca
Alice : Present
Bob : Absent
```

## Explanation:

- Dictionary stores student → attendance
- Loop marks multiple students
- Simple and efficient

## Task 5: ATM Menu System (Loops + Conditionals)

**Prompt:**

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

## Python code:

```python
balance = 10000

while True:
    print("\n1. Check Balance")
    print("2. Withdraw")
    print("3. Deposit")
    print("4. Exit")

    choice = int(input("Enter choice: "))

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

    elif choice == 2:
        amt = int(input("Enter amount: "))
        if amt <= balance:
            balance -= amt
            print("Withdrawal successful")
        else:
            print("Insufficient balance")

    elif choice == 3:
        amt = int(input("Enter amount: "))
        balance += amt
        print("Deposit successful")

    elif choice == 4:
        print("Thank you")
        break

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

## Output:

```
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter choice: 1
Balance: 10000

1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter choice: 1
Balance: 10000

1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter choice: 2
Enter amount: 5000
Withdrawal successful

1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter choice: 4
Thank you
```

## Explanation:

The program effectively simulates real-world ATM operations.

Conditional statements and loops are used correctly.

The system handles invalid inputs and balance conditions properly.