

# AI ASSISTED CODING

## LAB-6.5

Aashritha

2303A51756

Batch-11

### **Task Description #1 (AI-Based Code Completion for Conditional Eligibility Check)**

**Task:** Use an AI tool to generate eligibility logic.

**Prompt:**

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

**Expected Output:**

- AI-generated conditional logic.
- Correct eligibility decisions.
- Explanation of conditions.

**Prompt:**

```
# Give a python code to check whether a person is eligible to vote or not based on their age and citizenship
```

### **Given Code and Output:**

The screenshot shows a terminal window with the following content:

```
lab6.py > ...
1  # Give a python code to check whether a person is eligible to vote or not based on their age and citizenship
2  def is_eligible_to_vote(age, is_citizen):
3      if age >= 18 and is_citizen:
4          return True
5      else:
6          return False
7  # Example usage
8  age = int(input("Enter your age: "))
9  is_citizen = input("Are you a citizen? (yes/no): ").strip().lower() == 'yes'
10 if is_eligible_to_vote(age, is_citizen):
11     print("You are eligible to vote.")
12 else:
13     print("You are not eligible to vote.")

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\Users\katta\OneDrive\Desktop\AIAC> & C:/Users/katta/anaconda3/python.exe c:/users/katta/OneDrive/Desktop/AIAC/lab6.py
Enter your age: 20
Are you a citizen? (yes/no): yes
You are eligible to vote.
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

### **Task Description #2(AI-Based Code Completion for Loop-Based String Processing)**

**Task:** Use an AI tool to process strings using loops.

**Prompt:**

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

**Expected Output:**

- AI-generated string processing logic.
- Correct counts.
- Output verification.

**Prompt:**

```
#Given a python code to count vowels and consonants in a string using a loop
```

**Given Code and Output:**

```
.lab6.py > ...
1  #Given a python code to count vowels and consonants in a string using a loop
2  def count_vowels_consonants(input_string):
3      vowels = "aeiouAEIOU"
4      vowel_count = 0
5      consonant_count = 0
6
7      for char in input_string:
8          if char.isalpha(): # Check if the character is a letter
9              if char in vowels:
10                  vowel_count += 1
11              else:
12                  consonant_count += 1
13
14      return vowel_count, consonant_count
15 # Example usage
16 input_string = input("Enter a string:")
17 vowels, consonants = count_vowels_consonants(input_string)
18 print(f"Vowels: {vowels}, Consonants: {consonants}")
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\katta\OneDrive\Desktop\AIAC> & C:/Users/katta/anaconda3/python.exe c:/Users/katta/OneDrive/Desktop/AIAC/lab6.py
Enter a string:Hello World
Vowels: 3, Consonants: 7
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

**Task Description #3 (AI-Assisted Code Completion Reflection)**

**Task)**

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

**Prompt:**

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

**Expected Output:**

- Complete AI-generated program.
- Review of AI suggestions quality.
- Short reflection on AI-assisted coding experience.

**Prompt:**

#Give a python code for library management system using classes, loops and conditional statements

**Given Code:**

```

1 #Give a python code for library management system using classes, loops and conditional statements
2 class Book:
3     def __init__(self, title, author):
4         self.title = title
5         self.author = author
6         self.is_available = True
7 class Library:
8     def __init__(self):
9         self.books = []
10    def add_book(self, book):
11        self.books.append(book)
12        print(f'Book "{book.title}" by {book.author} added to the library.')
13    def display_books(self):
14        print("Available books in the library:")
15        for book in self.books:
16            status = "Available" if book.is_available else "Checked out"
17            print(f'Title: {book.title}, Author: {book.author}, Status: {status}')
18    def borrow_book(self, title):
19        for book in self.books:
20            if book.title == title:
21                if book.is_available:
22                    book.is_available = False
23                    print(f'You have borrowed "{book.title}"')
24                    return
25                else:
26                    print(f'Sorry, "{book.title}" is currently checked out.')
27                    return
28        print(f'Sorry, "{title}" is not available in the library.')
29    def return_book(self, title):
30        for book in self.books:
31            if book.title == title:
32                if not book.is_available:
33                    book.is_available = True
34                    print(f'You have returned "{book.title}"')
35                    return
36                else:
37                    print(f'{book.title} was not checked out.')
38                    return
39        print(f'Sorry, "{title}" does not belong to this library.')
40 def main():
41     library = Library()
42     while True:
43         print("\nLibrary Management System")
44         print("1. Add Book")
45         print("2. Display Books")
46         print("3. Borrow Book")
47         print("4. Return Book")
48         print("5. Exit")
49         choice = input("Enter your choice (1-5): ")
50         if choice == '1':
51             title = input("Enter book title: ")
52             author = input("Enter book author: ")
53             book = Book(title, author)
54             library.add_book(book)
55         elif choice == '2':
56             library.display_books()
57         elif choice == '3':
58             title = input("Enter the title of the book to borrow: ")
59             library.borrow_book(title)
60         elif choice == '4':
61             title = input("Enter the title of the book to return: ")
62             library.return_book(title)
63         elif choice == '5':
64             print("Exiting the Library Management System. Goodbye!")
65             break
66         else:
67             print("Invalid choice. Please try again.")
68 if __name__ == "__main__":
69     main()

```

### Output:

```
Library Management System
1. Add Book
2. Display Books
3. Borrow Book
4. Return Book
5. Exit
Enter your choice (1-5): 1
Enter book title: The Time Machine
Enter book author: H.G. Wells
Book "The Time Machine" by H.G. Wells added to the library.
```

```
Library Management System
1. Add Book
2. Display Books
3. Borrow Book
4. Return Book
5. Exit
Enter your choice (1-5): 2
Available books in the library:
Title: "The Time Machine", Author: H.G. Wells, Status: Available
```

```
Library Management System
1. Add Book
2. Display Books
3. Borrow Book
4. Return Book
5. Exit
Enter your choice (1-5): 3
Enter the title of the book to borrow: The Time Machine
You have borrowed "The Time Machine".
```

```
Library Management System
1. Add Book
2. Display Books
3. Borrow Book
4. Return Book
5. Exit
Enter your choice (1-5): 4
Enter the title of the book to return: The Time Machine
You have returned "The Time Machine".
```

#### Task Description #4 (AI-Assisted Code Completion for Class-Based Attendance System)

**Task:** Use an AI tool to generate an attendance management class.

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

**Expected Output:**

- AI-generated attendance logic.

- Correct display of attendance.

- Test cases.

**Prompt:**

# Give a python code to mark and display student attendance using loops.

**Given Code:**

```
# Give a python code to mark and display student attendance using loops.
def mark_attendance(students):
    attendance = {}
    for student in students:
        status = input(f"Is {student} present? (y/n): ").strip().lower()
        if status == 'y':
            attendance[student] = 'Present'
        else:
            attendance[student] = 'Absent'
    return attendance
def display_attendance(attendance):
    print("\nAttendance Record:")
    for student, status in attendance.items():
        print(f"{student}: {status}")
if __name__ == "__main__":
    student_list = ["Alice", "Bob", "Charlie", "David"]
    attendance_record = mark_attendance(student_list)
    display_attendance(attendance_record)
```

**Output:**

```
PS C:\Users\katta\OneDrive\Desktop\AIAC> & C:/Use
Is Alice present? (y/n): y
Is Bob present? (y/n): y
Is Charlie present? (y/n): n
Is David present? (y/n): y
```

Attendance Record:

Alice: Present

Bob: Present

Charlie: Absent

David: Present

```
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

**Task Description #5 (AI-Based Code Completion for Conditional Menu Navigation)**

**Task:** Use an AI tool to complete a navigation menu.

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

### **Expected Output:**

- AI-generated menu logic.
- Correct option handling.
- Output verification.

### **Prompt:**

# Give a python code using loops and conditional statements to create an ATM menu

### **Given Code:**

```
# Give a python code using loops and conditional statements to create an ATM menu
def atm_menu():
    balance = 1000 # Initial balance
    while True:
        print("\n--- ATM Menu ---")
        print("1. Check Balance")
        print("2. Deposit Money")
        print("3. Withdraw Money")
        print("4. Exit")

        choice = input("Please select an option (1-4): ")

        if choice == '1':
            print(f"Your current balance is: ${balance}")

        elif choice == '2':
            amount = float(input("Enter amount to deposit: $"))
            if amount > 0:
                balance += amount
                print(f"${amount} deposited successfully.")
            else:
                print("Invalid amount. Please enter a positive number.")

        elif choice == '3':
            amount = float(input("Enter amount to withdraw: $"))
            if 0 < amount <= balance:
                balance -= amount
                print(f"${amount} withdrawn successfully.")
            else:
                print("Invalid amount. Please check your balance and try again.")

        elif choice == '4':
            print("Thank you for using the ATM. Goodbye!")
            break

        else:
            print("Invalid selection. Please choose a valid option (1-4).")

# Run the ATM menu
atm_menu()
```

### **Output:**

```
PS C:\Users\katta\OneDrive\Desktop\AIAC> & C:/Users/katta/aiac/ATM.py

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 1
Your current balance is: $1000

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 2
Enter amount to deposit: $10000
$10000.0 deposited successfully.

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 3
Enter amount to withdraw: $100
$100.0 withdrawn successfully.

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 4
Thank you for using the ATM. Goodbye!
```