

# AI Assisted Coding

## Assignment-6.5

Name: Mohammed Rahil Raza

Ht.no:2303A51433

Bt.no:21

---

### 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."

Code:

```
C:\> Users\rahil\OneDrive\Documents > assign 6.5.py > ...
1  '''generate a python program that print code to check voting eligibility based on age and citizenship status without using functions
2  input must be taken from user
3  include if the age input is not numeric then print invalid age input use numeric value only
4  '''
5  age_input = input("Enter your age: ")
6  citizenship_status = input("Are you a citizen? (yes/no): ").strip().lower()
7  if not age_input.isdigit():
8      print("Invalid age input. Use numeric value only.")
9  else:
10     age = int(age_input)
11     if age >= 18 and citizenship_status == 'yes':
12         print("You are eligible to vote.")
13     else:
14         print("You are not eligible to vote.")
15
```

Output:

```
Enter your age: 12
Are you a citizen? (yes/no): yes
You are not eligible to vote.
PS C:\Users\rahil\OneDrive\Documents> 
```

### 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.”

**Code:**

```
C:\> Users > rahil > OneDrive > Documents > assign 6.5.py > ...
1
2 ''' generate a python program that prints code to count and consonant in a string using a loop
3 check output verification
4 and generate string processing logic '''
5 input_string = input("Enter a string: ")
6 vowel_count = 0
7 consonant_count = 0
8 vowels = "aeiouAEIOU"
9 for char in input_string:
10     if char.isalpha():
11         if char in vowels:
12             vowel_count += 1
13         else:
14             consonant_count += 1
15 print(f"Vowels: {vowel_count}")
16 print(f"Consonants: {consonant_count}")
17
```

**Output:**

```
Enter a string: Rahil
Vowels: 2
Consonants: 3
```

### **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.

Code:

```
''' generate a python program using class loops and conditionals to create a library management system
'''
class Library:
    def __init__(self):
        self.books = {}

    def add_book(self, title, author):
        self.books[title] = author
        print(f'Book "{title}" by {author} added to the library.')

    def remove_book(self, title):
        if title in self.books:
            del self.books[title]
            print(f'Book "{title}" removed from the library.')
        else:
            print(f'Book "{title}" not found in the library.')

    def display_books(self):
        if not self.books:
            print("No books in the library.")
        else:
            print("Books in the library:")
            for title, author in self.books.items():
                print(f'"{title}" by {author}')

library = Library()
while True:
    print("\nLibrary Management System")
    print("1. Add Book")
    print("2. Remove Book")
    print("3. Display Books")
    print("4. Exit")
    choice = input("Enter your choice (1-4): ")
    if choice == '1':
        title = input("Enter book title: ")
        author = input("Enter book author: ")
        library.add_book(title, author)
    elif choice == '2':
        title = input("Enter book title to remove: ")
        library.remove_book(title)
    elif choice == '3':
        library.display_books()
    elif choice == '4':
        print("Exiting the Library Management System.")
        break
    else:
        print("Invalid choice. Please enter a number between 1 and 4.")
```

```
Enter your choice (1-4): 1
Enter book title: ikagi
Enter book author: idk
Book "ikagi" by idk added to the library.

Library Management System
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice (1-4): 3
Books in the library:
"ikagi" by idk
```

Output:

#### **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.

Code:

```
''' generate a python program attendance management
class to mark and display student attendance using loops
using test cases to verify output correctness '''
class Attendance:
    def __init__(self):
        self.attendance_record = {}

    def mark_attendance(self, student_name, present):
        self.attendance_record[student_name] = present

    def display_attendance(self):
        print("Attendance Record:")
        for student, present in self.attendance_record.items():
            status = "Present" if present else "Absent"
            print(f"{student}: {status}")
attendance = Attendance()
students = ["Alice", "Bob", "Charlie", "David"]
for student in students:
    status_input = input(f"Is {student} present? (yes/no): ").strip().lower()
    present = True if status_input == 'yes' else False
    attendance.mark_attendance(student, present)
attendance.display_attendance()
```

Output:

```
Is Alice present? (yes/no): yes
Is Bob present? (yes/no): yes
Is Charlie present? (yes/no): no
Is David present? (yes/no): yes
Attendance Record:
Alice: Present
Bob: Present
Charlie: Absent
David: Present
```

### **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

Code:

```
''' generate a python program that completes a navigation menu to simulate an ATM menu
output' verification using conditionals and loops '''
balance = 1000.0
while True:
    print("\nATM Menu")
    print("1. Check Balance")
    print("2. Deposit Money")
    print("3. Withdraw Money")
    print("4. Exit")
    choice = input("Enter your choice (1-4): ")
    if choice == '1':
        print(f"Your current balance is: ${balance:.2f}")
    elif choice == '2':
        amount = float(input("Enter amount to deposit: "))
        if amount > 0:
            balance += amount
            print(f"${amount:.2f} deposited successfully.")
        else:
            print("Invalid amount. Please enter a positive value.")
    elif choice == '3':
        amount = float(input("Enter amount to withdraw: "))
        if 0 < amount <= balance:
            balance -= amount
            print(f"${amount:.2f} withdrawn successfully.")
        else:
            print("Invalid amount or insufficient balance.")
    elif choice == '4':
        print("Exiting the ATM. Thank you!")
        break
    else:
        print("Invalid choice. Please enter a number between 1 and 4.")
```

Output:

ATM Menu

1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit

Enter your choice (1-4): 2

Enter amount to deposit: 1000

\$1000.00 deposited successfully.

ATM Menu

1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit

Enter your choice (1-4): 3

Enter amount to withdraw: 500

\$500.00 withdrawn successfully.