Assignment -6.5

Name: G. Dhanush Reddy

Ht.no:2303A51619 **Bt.no:22**

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:

```
'''Task Description #1 (AI-Based Code Completion for Conditional Eligibility Check)
Task: Use an AI tool to generate eligibility logic.
“Generate Python code to check voting eligibility based on age and citizenship.
'''
def is_eligible_to_vote(age, is_citizen):
    """
    Check if a person is eligible to vote based on age and citizenship status
    :param age: int - Age of the person
    :param is_citizen: bool - Citizenship status of the person
    :return: bool - True if eligible to vote, False otherwise
    """
    if age >= 18 and is_citizen:
        return True
    else:
        return False

#i should give dyanamically age and is_citizen values to the function is_eligible_to_vote enter citizedn status as yes or no
age = int(input("Enter your age: "))
citizenship_input = input("Are you a citizen:").strip().lower()
is_citizen = citizenship_input == 'indian'
eligibility = is_eligible_to_vote(age, is_citizen)
print(f"Eligible to vote: {eligibility}")
```

Output:

```
Enter your age: 44
Are you a citizen:indian
Eligible to vote: True
```

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:

```
'''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.
output should be printed below'''
''
def count_vowels_and_consonants(input_string):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0

    for char in input_string:
        if char.isalpha():  # Check if the character is a letter
            if char in vowels:
                vowel_count += 1
            else:
                consonant_count += 1

    return vowel_count, consonant_count
```

Output:

```
Enter a string: Dhanush reddy
Vowels: 3, Consonants: 9
PS C:\Users\Dhanush Reddy>
```

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:

```
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author
        self.is_available = True

    def __str__(self):
        return f"{self.title} by {self.author} - {'Available' if self.is_available else 'Checked Out'}"
class Library:
    def __init__(self):
        self.books = []

    def add_book(self, book):
        self.books.append(book)
        print(f'Added: {book}')

    def display_books(self):
        print("Library Catalog:")
        for book in self.books:
            print(book)

    def check_out_book(self, title):
        for book in self.books:
            if book.title == title:
                if book.is_available:
                    book.is_available = False
                    print(f'Checked out: {book}')
                    return
                else:
                    print(f'Sorry, {book.title} is currently checked out.')
                    return
        print(f'Sorry, {title} is not in the catalog.')

    def return_book(self, title):
        for book in self.books:
            if book.title == title:
                if not book.is_available:
                    book.is_available = True
                    print(f'Returned: {book}')
                    return
                else:
                    print(f'{book.title} was not checked out.')
                    return
        print(f'Sorry, {title} is not in the catalog.')
# Example Usage
if __name__ == "__main__":
    library = Library()
    library.add_book(Book("1984", "George Orwell"))
    library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
    library.add_book(Book("The Great Gatsby", "F. Scott Fitzgerald"))

    library.display_books()

    library.check_out_book("1984")
    library.check_out_book("The Great Gatsby")
    library.display_books()
```

Output:

```
Added: 1984 by George Orwell - Available
Added: To Kill a Mockingbird by Harper Lee - Available
Added: The Great Gatsby by F. Scott Fitzgerald - Available
Library Catalog:
1984 by George Orwell - Available
To Kill a Mockingbird by Harper Lee - Available
The Great Gatsby by F. Scott Fitzgerald - Available
Checked out: 1984 by George Orwell - Checked Out
Checked out: The Great Gatsby by F. Scott Fitzgerald - Checked Out
Library Catalog:
To Kill a Mockingbird by Harper Lee - Available
The Great Gatsby by F. Scott Fitzgerald - Available
Checked out: 1984 by George Orwell - Checked Out
Checked out: The Great Gatsby by F. Scott Fitzgerald - Checked Out
Library Catalog:
Checked out: 1984 by George Orwell - Checked Out
Checked out: The Great Gatsby by F. Scott Fitzgerald - Checked Out
Library Catalog:
Checked out: The Great Gatsby by F. Scott Fitzgerald - Checked Out
Library Catalog:
Library Catalog:
1984 by George Orwell - Checked Out
To Kill a Mockingbird by Harper Lee - Available
The Great Gatsby by F. Scott Fitzgerald - Checked Out
Returned: 1984 by George Orwell - Available
Library Catalog:
1984 by George Orwell - Available
To Kill a Mockingbird by Harper Lee - Available
The Great Gatsby by F. Scott Fitzgerald - Checked Out
```

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:

```
class AttendanceSystem:
    def __init__(self):
        self.attendance = {}

    def mark_attendance(self, student_name):
        """Marks attendance for a given student."""
        self.attendance[student_name] = 'Present'

    def display_attendance(self):
        """Displays the attendance of all students."""
        print("Attendance Record:")
        for student, status in self.attendance.items():
            print(f"{student}: {status}")
# Test cases
if __name__ == "__main__":
    attendance_system = AttendanceSystem()

    # Mark attendance for students
    attendance_system.mark_attendance("Alice")
    attendance_system.mark_attendance("Bob")
    attendance_system.mark_attendance("Charlie")

    # Display attendance
    attendance_system.display_attendance()
    # Expected Output:
    # Attendance Record:
    # Alice: Present
    # Bob: Present

    # Charlie: Present
```

Output:

```
Attendance Record:
Alice: Present
Bob: Present
Charlie: 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:

```
def atm_menu():
    balance = 1000  # Initial balance
    while True:
        print("\nWelcome to the ATM")
        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':
            deposit_amount = float(input("Enter amount to deposit: $"))
            if deposit_amount > 0:
                balance += deposit_amount
                print(f"${deposit_amount} deposited successfully.")
            else:
                print("Invalid amount. Please try again.")

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

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

        else:
            print("Invalid option. Please select a valid option (1-4).")
# Run the ATM menu
atm_menu()
```

Output:

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

Welcome to the ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4):
```