

School of Computer Science and Artificial Intelligence

Lab Assignment # 6

Name of Student : GINNE SAI TEJA
Enrollment No. : 2303A51526
Batch No. : 22

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

Task: Use an AI tool to generate eligibility logic.

Prompt:

“write a check eligibility for voting based on age and citizenship.”

CODE:

```
#task-1
#write a check eligibility for voting based on age and citizenship
age = int(input("Enter your age: "))
citizenship = input("Are you a citizen? (yes/no): ").strip().lower()
if age >= 18 and citizenship == 'yes':
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote.")
```

OUTPUT:

```
Enter your age: 16
Are you a citizen? (yes/no): yes
You are not eligible to vote.
PS C:\AIAC-LAB> ^C
PS C:\AIAC-LAB>
PS C:\AIAC-LAB> c:; cd 'c:\AIAC-LAB'
hon.debugpy-2025.19.2026012701-win32-
Enter your age: 20
Are you a citizen? (yes/no): yes
You are eligible to vote.
PS C:\AIAC-LAB> □
```

Justification: The code checks if the user is 18 years or older and if they are a citizen. If both conditions are met, it prints that the user is eligible to vote; otherwise, it states they are not eligible.

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

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

Prompt: write a program to count vowels and consonants in a given string

Code:

```
#task-2
#write a program to count vowels and consonants in a given string
string = input("Enter a string: ")
vowels = "aeiouAEIOU"
vowel_count = 0
consonant_count = 0
for char in string:
    if char.isalpha():
        if char in vowels:
            vowel_count += 1
        else:
            consonant_count += 1
print(f"Vowels: {vowel_count}, Consonants: {consonant_count}")
```

Output:

```
Enter a string: sai teja
Vowels: 4, Consonants: 3
PS C:\AIAC-LAB>
```

Justification:

→The first task checks if a person is eligible to vote based on their age and citizenship status. It uses conditional statements to determine eligibility.

→The second task counts the number of vowels and consonants in a given string by iterating through each character and checking if it is a vowel or consonant.

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: write a program for library management system using classes, loops and conditional statements

Output:

```
#task-3
#write a program for library management system using classes, loops and conditional statements
class Library:
    def __init__(self):
        self.books = []

    def add_book(self, book):
        self.books.append(book)
        print(f'Book "{book}" added to the library.')

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

    def display_books(self):
        if self.books:
            print("Books in the library:")
            for book in self.books:
                print(f'- {book}')
        else:
            print("No books in the library.")

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':
        book = input("Enter the book name to add: ")
        library.add_book(book)
    elif choice == '2':
        book = input("Enter the book name to remove: ")
```

```
    elif choice == '2':
        book = input("Enter the book name to remove: ")
        library.remove_book(book)
    elif choice == '3':
        library.display_books()
    elif choice == '4':
        print("Exiting the program.")
        break
    else:
        print("Invalid choice. Please try again.")
```

Output:

```
2. Remove Book
3. Display Books
4. Exit
Enter your choice (1-4): 1
Enter the book name to add: HPC
Book "HPC" 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:
- AIAC
- CP
- HPC

Library Management System
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice (1-4): 2
Enter the book name to remove: HPC
Book "HPC" removed from the library.

Library Management System
1. Add Book
2. Remove Book
3. Display Books
4. Exit
Enter your choice (1-4): 4
Exiting the program.
PS C:\AIAC-LAB>
```

Justification:

→ The Library class encapsulates the functionality of a library management system.

→ It includes methods to add, remove, and display books.

→ A while loop is used to continuously prompt the user for actions until they choose to exit.

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

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

Prompt: Write a python program class to mark attendance of students in a class using class, loops and conditional statements

Code:

```
#Task-4
#Write a python program class to mark attendance of students in a class using class, loops and conditional statements
class Attendance:
    def __init__(self):
        self.attendance_record = {}

    def mark_attendance(self, student_name, present):
        self.attendance_record[student_name] = present
        status = "present" if present else "absent"
        print(f"{student_name} marked as {status}.")

    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()
while True:
    print("\nAttendance Management System")
    print("1. Mark Attendance")
    print("2. Display Attendance")
    print("3. Exit")
    choice = input("Enter your choice (1-3): ")
    if choice == '1':
        student_name = input("Enter the student's name: ")
        present_input = input("Is the student present? (yes/no): ").strip().lower()
        present = True if present_input == 'yes' else False
        attendance.mark_attendance(student_name, present)
    elif choice == '2':
        attendance.display_attendance()
    elif choice == '3':
        print("Exiting the program.")
        break
    else:
        print("Invalid choice. Please try again.")
```

Justification:

→ This program uses a class to encapsulate attendance functionality, loops to allow multiple operations,

→ and conditional statements to handle user input and attendance status.

Output:

```
Attendance Management System
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice (1-3): 1
Enter the student's name: sagar
Is the student present? (yes/no): no
sagar marked as absent.

Attendance Management System
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice (1-3): 1
Enter the student's name: shiva
Is the student present? (yes/no): yes
shiva marked as present.

Attendance Management System
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice (1-3): 2
Attendance Record:
sai teja: Present
phanikumar: Present
sagar: Absent
shiva: Present

Attendance Management System
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice (1-3): 3
Exiting the program.
PS C:\AIAC-LAB>
```

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

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

Prompt: write a program using loops and conditional statements to simulate an atm menu(ai tools basic atm functions)

Code:

```
#Task 5
#Write a program using loops and conditional statements to simulate an atm menu(ai tools basic atm functions)
class ATM:
    def __init__(self, balance=0):
        self.balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited: ${amount:.2f}")
        else:
            print("Deposit amount must be positive.")

    def withdraw(self, amount):
        if amount > 0:
            if amount <= self.balance:
                self.balance -= amount
                print(f"Withdrew: ${amount:.2f}")
            else:
                print("Insufficient funds.")
        else:
            print("Withdrawal amount must be positive.")

    def check_balance(self):
        print(f"Current balance: ${self.balance:.2f}")

atm = ATM()
while True:
    print("\nATM Menu")
    print("1. Deposit")
    print("2. Withdraw")
    print("3. Check Balance")
    print("4. Exit")
    choice = input("Enter your choice (1-4): ")
    if choice == '1':
        amount = float(input("Enter amount to deposit: "))
        atm.deposit(amount)
    elif choice == '2':
        amount = float(input("Enter amount to withdraw: "))
        atm.withdraw(amount)
    elif choice == '3':
        atm.check_balance()
    elif choice == '4':
        print("Exiting the ATM. Thank you!")
        break
    else:
        print("Invalid choice. Please try again.")
```

Output:

```
4. Exit
Enter your choice (1-4): 1
Enter amount to deposit: 4000
Deposited: $4000.00
```

```
ATM Menu
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice (1-4): 1
Enter amount to deposit: 600
Deposited: $600.00
```

```
ATM Menu
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice (1-4): 2
Enter amount to withdraw: 300
Withdrew: $300.00
```

```
ATM Menu
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice (1-4): 3
Current balance: $4300.00
```

```
ATM Menu
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice (1-4): 4
Exiting the ATM. Thank you!
```

Justification:

- The program simulates an ATM menu with basic functions such as deposit, withdraw, and check balance.
- It uses a class to encapsulate the ATM functionality and employs loops and conditional statements to interact with the user.