

## School of Computer Science and Artificial Intelligence

---

**Lab Assignment # 6.5**

---

<b>Program</b>	: B. Tech (CSE)
<b>Specialization</b>	:
<b>Course Title</b>	: AI Assisted coding
<b>Course Code</b>	:
<b>Semester</b>	: II
<b>Academic Session</b>	: 2025-2026
<b>Name of Student</b>	:Ganesh
<b>Enrollment No.</b>	: 2403A51L55
<b>Batch No.</b>	: 52
<b>Date</b>	:22-01-2026

---

**Task 1: Use an AI tool to generate eligibility logic.**

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

**Code :**

```
▶ age = int(input("Enter age: "))
    citizen = input("Citizen (yes/no): ")
    if age >= 18 and citizen == "yes":
        print("Eligible to vote")
    else:
        print("Not eligible")
```

**Output:**

```
... Enter age: 18
    Citizen (yes/no): yes
    Eligible to vote
```

**Explanation :**

- The program takes age as integer input from the user.
- It takes citizenship status as string input (yes or no).
- The if statement checks two conditions together.
- First condition verifies whether age is 18 or above.
- Second condition checks if the user is a citizen.

- The and operator ensures both conditions must be true.
- If both are true, voting eligibility is confirmed.
- The program prints "Eligible to vote".
- If any condition fails, control goes to else.
- It prints "Not eligible" when criteria are not met.

### Task 2 : Count Vowels and Consonants (Loop + String)

Code :

```
▶ s = input("Enter string: ").lower()
v = c = 0
for ch in s:
    if ch.isalpha():
        if ch in "aeiou":
            v += 1
        else:
            c += 1
print("Vowels:", v, "Consonants:", c)
|
```

Output :

```
... Enter string: kaizen
Vowels: 3 Consonants: 3
```

Explanation :

- The program takes a string input from the user.
- The string is converted to lowercase for easy comparison.
- Two variables v and c are initialized to count vowels and consonants.
- A for loop iterates through each character in the string.
- isalpha() checks whether the character is a letter.
- Non-alphabet characters like spaces and numbers are ignored.
- The if condition checks if the character is a vowel.
- Vowels are counted using variable v.
- Remaining letters are counted as consonants using c.
- The final count of vowels and consonants is displayed.

**Task 3: Library Management System (Class + Loop + Condition)**

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

**Code :**

```
① class Library:  
    def __init__(self):  
        self.books = []  
    def add_book(self, b):  
        self.books.append(b)  
    def show_books(self):  
        for b in self.books:  
            print(b)  
  
lib = Library()  
while True:  
    ch = input("1.Add 2.Show 3.Exit: ")  
    if ch == "1": lib.add_book(input("Book name: "))  
    elif ch == "2": lib.show_books()  
    else: break
```

**Output :**

```
... 1.Add 2.Show 3.Exit: 1  
Book name: kaizen
```

**Explanation :**

- The program defines a class named **Library**.
- The constructor initializes an empty list to store books.
- **add\_book()** method adds a new book to the list.
- **show\_books()** method displays all stored books.
- An object **lib** is created from the **Library** class.
- A while True loop keeps the program running continuously.
- The user is shown a menu with add, show, and exit options.
- If choice is 1, a book name is added to the library.
- If choice is 2, all books are displayed.
- Any other choice exits the program.

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

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

**Code:**

```
▶ class Attendance:  
    def __init__(self):  
        self.data = {}  
    def mark(self, name):  
        self.data[name] = "Present"  
    def display(self):  
        for k, v in self.data.items():  
            print(k, ":", v)  
  
a = Attendance()  
a.mark("Sai")  
a.mark("Akash")  
a.display()
```

**Output :**

```
...  Sai : Present  
      Akash : Present
```

**Explanation :**

- The program defines a class named Attendance.
- The constructor initializes an empty dictionary to store attendance data.
- The dictionary stores student names as keys.
- Attendance status is stored as values in the dictionary.
- The mark() method marks a student as "Present".
- The display() method prints all attendance records.
- A for loop is used to access dictionary items.
- An object a is created from the Attendance class.
- Attendance is marked for students Sai and Ravi.
- The final attendance list is displayed on the screen.

## Task 5: ATM Menu Simulation (Loop + Conditionals)

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

```
▶ bal = 1000
  while True:
      ch = input("1.Deposit 2.Withdraw 3.Balance 4.Exit: ")
      if ch == "1": bal += int(input("Amount: "))
      elif ch == "2": bal -= int(input("Amount: "))
      elif ch == "3": print("Balance:", bal)
      else: break
```

**Output :**

```
... 1.Deposit 2.Withdraw 3.Balance 4.Exit: 1
    Amount: 5500
    1.Deposit 2.Withdraw 3.Balance 4.Exit: 2
    Amount: 2000
    1.Deposit 2.Withdraw 3.Balance 4.Exit: 3
    Balance: 4500
    1.Deposit 2.Withdraw 3.Balance 4.Exit: 4
```

**Explanation :**

- ▀ The program initializes the account balance with 1000.
- ▀ A while True loop keeps the ATM menu running.
- ▀ The user is shown options for deposit, withdraw, balance, and exit.
- ▀ User input is stored in the variable ch.
- ▀ If the choice is 1, the entered amount is added to the balance.
- ▀ If the choice is 2, the entered amount is subtracted from the balance.
- ▀ If the choice is 3, the current balance is displayed.
- ▀ Conditional statements control the menu flow.
- ▀ The loop repeats after each operation.
- ▀ The program stops when the user selects exit.