SR UNIVERSITY

School of Computer Science and Artificial Intelligence

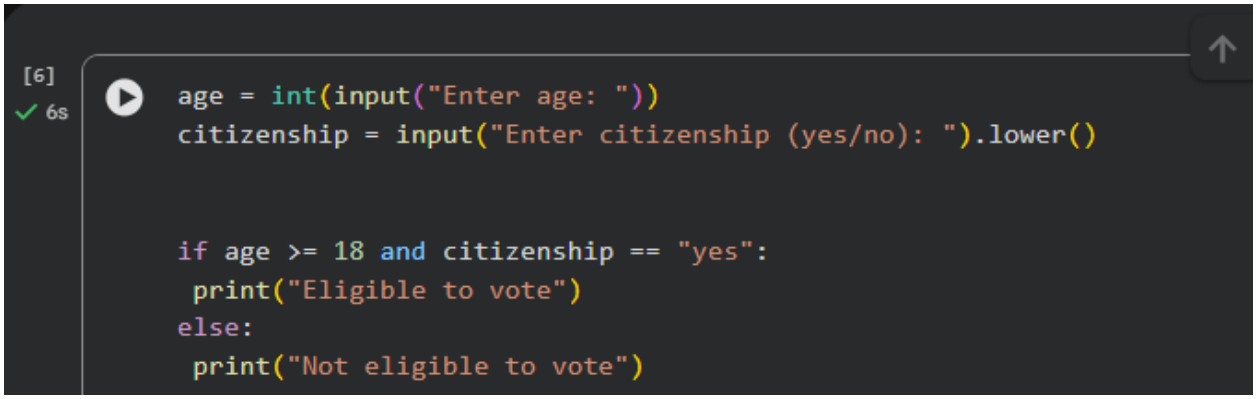
Lab Assignment # 6.5

Program	**: B. Tech (CSE)**
Specialization	**:**
Course Title	**: AI Assisted coding**
Course Code	**:**
Semester	**: II**
Academic Session	**: 2025-2026**
Name of Student	**: M.Srinath**
Enrollment No.	**: 2403A51L29**
Batch No.	**: 51**
Date	**:23-01-2026**

Task-1

Prompt

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



Explanation:

`age = int(input(...))` takes the user's age as an integer.

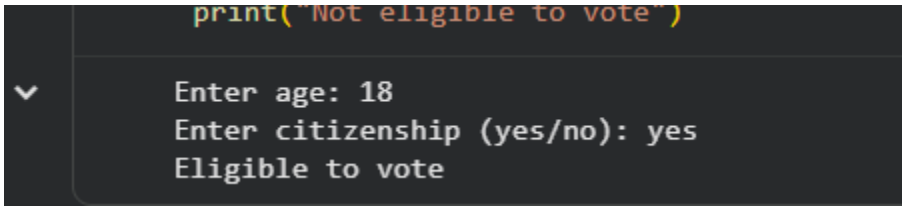
`citizenship = input(...).lower()` reads citizenship status and converts it to lowercase for comparison.

`if age >= 18 and citizenship == "yes":` checks both conditions together.

`print("Eligible to vote")` executes when both conditions are true.

`else:` handles cases where one or both conditions fail.

Output:



Task-2

Prompt:

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

SR UNIVERSITY

```
[9]
✓ 9s
text = input("Enter a string: ")
vowels = "aeiou"
vowel_count = 0
consonant_count = 0


for ch in text.lower():
 if ch.isalpha():
  if ch in vowels:
    vowel_count += 1
else:
  consonant_count += 1


print("Vowels:", vowel_count)
print("Consonants:", consonant_count)
```

Explanation

- `text = input(...)` **reads the input string.**
- `vowels = "aeiou"` **stores all vowel characters.**
- **Counters are initialized to zero.**
- `for ch in text.lower():` **iterates through each character.**
- `ch.isalpha()` **ensures only letters are counted.**
- **Inner condition checks vowel or consonant.**
- **Final print statements display the counts.**

Output:

```
Enter a string:  hello world
Vowels: 3
Consonants: 1
```

Task-3

SR UNIVERSITY

Prompt:

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

Explanation

- `Library` **class stores book details.**
- **List** `books` **is used to manage book records.**
- `add_book()` **adds a new book to the list.**
- `display_books()` **uses a loop to display all books.**
- `while True` **loop allows repeated user interaction.**
- `if-elif-else` **handles menu selection using conditionals.**

Output:

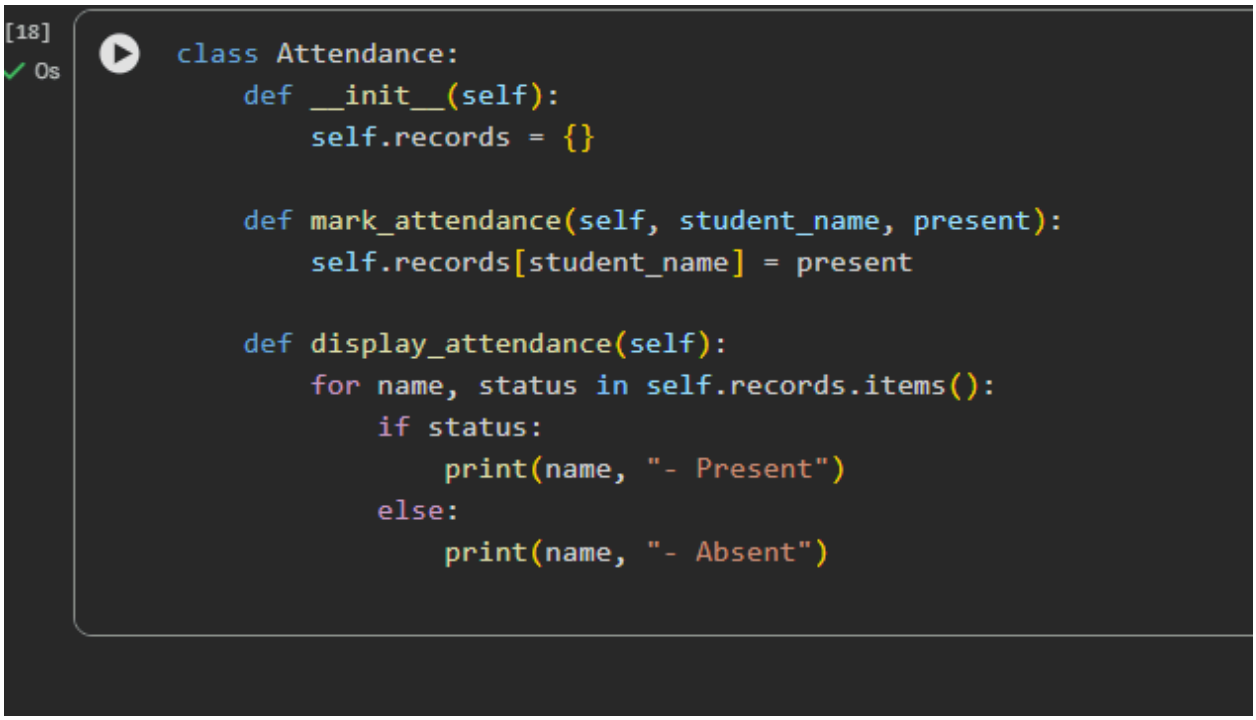
```
...
    1. Add Book
    2. Display Books
    3. Exit
    Enter choice: 1
    Enter book name: python

    1. Add Book
    2. Display Books
    3. Exit
    Enter choice: 1
    Enter book name: java

    1. Add Book
    2. Display Books
    3. Exit
    Enter choice: 2
    python
    java

    1. Add Book
    2. Display Books
    3. Exit
    Enter choice: 3
```

SR UNIVERSITY

Task-4:

Prompt Used:

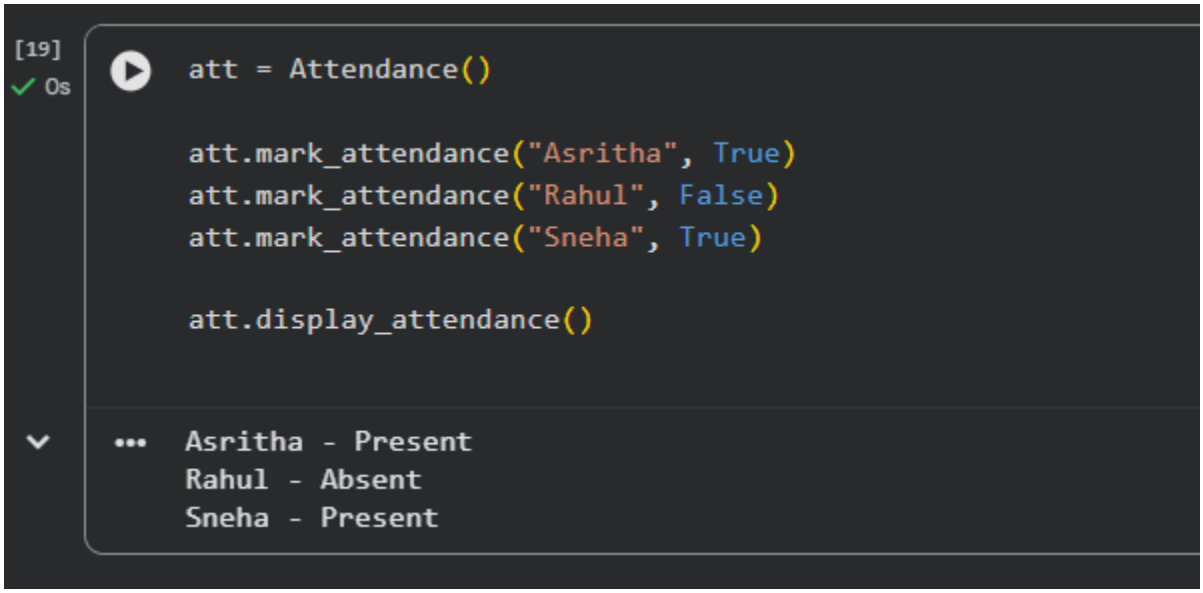
Generate a Python class to mark and display student attendance using loops.



Explanation

- `Attendance` **class stores attendance details.**
- **Dictionary** `records` **stores student names and their attendance status.**
- `mark_attendance()` **records whether a student is present or absent.**
- `display_attendance()` **uses a loop to display attendance for all students.**

Output:



Task-5

Prompt:

.Generate a Python program using loops and conditionals to simulate an ATM menu.

SR UNIVERSITY

[17] 1m

```python
balance = 5000

while True:
    print("\nATM MENU")
    print("1. Check Balance")
    print("2. Deposit")
    print("3. Withdraw")
    print("4. Exit")

    choice = int(input("Enter your choice: "))

    if choice == 1:
        print("Current Balance:", balance)

    elif choice == 2:
        amount = int(input("Enter deposit amount: "))
        if amount > 0:
            balance += amount
            print("Amount deposited successfully")
        else:
            print("Invalid amount")

    elif choice == 3:
        amount = int(input("Enter withdrawal amount: "))
        if amount <= balance and amount > 0:
            balance -= amount
            print("Please collect your cash")
        else:
            print("Insufficient balance or invalid amount")

    elif choice == 4:
        print("Thank you for using ATM")
        break

    else:
        print("Invalid option")
```

SR UNIVERSITY

Explanation

- `while True` **creates a continuous menu loop.**
- **Menu options are displayed using** `print`**.**
- `if-elif-else` **handles user selections.**
- **Balance is updated using conditional checks.**
- **Loop exits when the user selects option 4.**

SR UNIVERSITY

Output:

```
...
    ATM MENU
    1. Check Balance
    2. Deposit
    3. Withdraw
    4. Exit
    Enter your choice: 1
    Current Balance: 5000

    ATM MENU
    1. Check Balance
    2. Deposit
    3. Withdraw
    4. Exit
    Enter your choice: 2
    Enter deposit amount: 5000
    Amount deposited successfully

    ATM MENU
    1. Check Balance
    2. Deposit
    3. Withdraw
    4. Exit
    Enter your choice: 3
    Enter withdrawal amount: 2000
    Please collect your cash

    ATM MENU
    1. Check Balance
    2. Deposit
    3. Withdraw
    4. Exit
    Enter your choice: 4
    Thank you for using ATM
```

SR UNIVERSITY

Conclusion

In this experiment, AI-based code completion was used to generate Python programs for all five tasks involving conditionals, loops, classes, and menu-driven logic. Each task helped in understanding how AI can assist in writing correct and structured code. By reviewing outputs and making improvements, the importance of testing and ethical use of AI tools was learned.