

School of Computer Science and Artificial Intelligence

Lab Assignment 6.5

Program : B. Tech (CSE)
Course Title : AI Assisted Coding
Course Code : 23CS002PC304
Semester : III
Academic Session : 2025-2026
Name of Student : Sruthi
Enrollment No. : 2403A51L10
Batch No. : 51
Date : 23-01-2026

Task 1:

Prompt: Write a python program to check voting eligibility based on age and citizenship.

Code:

```
def check_voting_eligibility(age, is_citizen):
    if age >= 18 and is_citizen:
        return "Eligible to vote"
    else:
        return "Not eligible to vote"
# Example usage
age = 20
is_citizen = True
result = check_voting_eligibility(age, is_citizen)
print(result)
```

Output:

```
| PS C:\Users\BURRA SRUTHI\OneDrive\Desktop\3_2\AI\AI_code> & "C:\Users\BURRA SRUTHI\OneDrive\Desktop\3_2\AI\AI_code\.venv\Scripts\python.exe" "c:/Users/BURRA SRUTHI/OneDrive/Desktop/3_2/AI/AI_code/Assign_6-5.py"
| Eligible to vote
```

Explanation:

The function checks if the age is 18 or older and if the person is a citizen. If both conditions are met, it returns that the person is eligible to vote; otherwise, it states they are not eligible.

Task 2:

Prompt: Write the python code to count vowels and consonants in a string using a loop.

Code:

```
def count_vowels_and_consonants(input_string):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0
    for char in input_string:
        if char.isalpha():
            if char in vowels:
                vowel_count += 1
            else:
                consonant_count += 1
    return vowel_count, consonant_count

# Example usage
input_string = "Hello World"
vowels, consonants = count_vowels_and_consonants(input_string)
print(f"Vowels: {vowels}, Consonants: {consonants}")
```

Output:

```
(.venv) PS C:\Users\BURRA SRUTHI\OneDrive\Desktop\3_2\AI\AI_code> & "C:\Users\BURRA SRUTHI\OneDrive\Desktop\3_2\AI\AI_code\.venv\Scripts\python.exe" "c:/Users/BURRA SRUTHI/OneDrive/Desktop/3_2/AI/AI_code/Assign_6-5.py"
Vowels: 3, Consonants: 7
```

Explanation:

The program defines a function that iterates through each character in the input string, checks if it's an alphabet letter, and then determines if it's a vowel or consonant to update the respective counts.

Task3:

Prompt: Write a python program for a library management system using classes, loops, and conditionals.

Code:

```

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

class Library:
    def __init__(self):
        self.books = []
    def add_book(self, book):
        self.books.append(book)
    def display_books(self):
        for book in self.books:
            status = "Available" if book.is_available else "Checked Out"
            print(f"Title: {book.title}, Author: {book.author}, Status: {status}")
    def check_out_book(self, title):
        for book in self.books:
            if book.title == title:
                if book.is_available:
                    book.is_available = False
                    return f"You have checked out '{title}'."
                else:
                    return f"'{title}' is currently not available."
        return f"'{title}' not found in the library."
    def return_book(self, title):
        for book in self.books:
            if book.title == title:
                book.is_available = True
                return f"You have returned '{title}'."
        return f"'{title}' not found in the library."
# Example usage
library = Library()
book1 = Book("1984", "George Orwell")
book2 = Book("To Kill a Mockingbird", "Harper Lee")
library.add_book(book1)
library.add_book(book2)
library.display_books()
print(library.check_out_book("1984"))
library.display_books()
print(library.return_book("1984"))
library.display_books()

```

Output:

```

(.venv) PS C:\Users\BURRA SRUTHI\OneDrive\Desktop\3_2\AI\AI_code> & "C:\Users\BURRA SRUTHI\OneDrive\Desktop\3_2\AI\AI_code\.venv\Scripts\python.exe" "c:/Users/BURRA SRUTHI/OneDrive/Desktop/3_2/AI/AI_code/Assign_6-5.py"
Title: 1984, Author: George Orwell, Status: Available
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available
You have checked out '1984'.
Title: 1984, Author: George Orwell, Status: Checked Out
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available
You have returned '1984'.
Title: 1984, Author: George Orwell, Status: Available
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available

```

Explanation:

This program defines a simple library management system with classes for Book and Library. It allows adding books, displaying their status, checking out, and returning books using loops and conditionals.

Task4:

Prompt: Write a python class to mark and display student attendance using loops.

Code:

```
class Student:  
    def __init__(self, name):  
        self.name = name  
        self.attendance_record = []  
    def mark_attendance(self, days_present):  
        for day in range(1, days_present + 1):  
            self.attendance_record.append(f"Day {day}: Present")  
    def display_attendance(self):  
        print(f"Attendance record for {self.name}:")  
        for record in self.attendance_record:  
            print(record)  
# Example usage  
student = Student("Alice")  
student.mark_attendance(5)  
student.display_attendance()
```

Output:

```
(.venv) PS C:\Users\BURRA SRUTHI\OneDrive\Desktop\3_2\AI\AI_code> & "C:\Users\BURRA SRUTHI\OneDrive\Desktop\3_2\AI\AI_code\.venv\Scripts\python.exe" "c:/Users/BURRA SRUTHI/OneDrive/Desktop/3_2/AI/AI_code/Assign_6-5.py"  
Attendance record for Alice:  
Day 1: Present  
Day 2: Present  
Day 3: Present  
Day 4: Present  
Day 5: Present
```

Explanation:

This program defines a Student class that can mark and display attendance for a student. The mark_attendance method records the days the student was present, and the display_attendance method prints the attendance record.

Task 5:

Prompt: Write a python program using loops and conditionals to simulate an ATM menu.

Code:

```
def atm_menu():
    balance = 1000 # Initial balance
    while True:
        print("\nATM Menu:")
        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':
            amount = float(input("Enter amount to deposit: $"))
            if amount > 0:
                balance += amount
                print(f"${amount} 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} withdrawn successfully.")
            else:
                print("Invalid amount or insufficient funds.")
        elif choice == '4':
            print("Thank you for using the ATM. Goodbye!")
            break
        else:
            print("Invalid selection. Please choose a valid option.")

# Start the ATM menu
atm_menu()
```

Output:

```
(.venv) PS C:\Users\BURRA SRUTHI\OneDrive\Desktop\3_2\AI\AI_code> & "C:\Users\BURRA SRUTHI\OneDrive\Desktop\3_2\AI\AI_code\.venv\Scripts\python.exe" "c:/Users/BURRA SRUTHI/OneDrive/Desktop/3_2/AI/AI_code/Assign_6-5.py"

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

ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 2
Enter amount to deposit: $1000
$1000.0 deposited successfully.

ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 3
Enter amount to withdraw: $500
$500.0 withdrawn successfully.

ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 1
Your current balance is: $1500.0

ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Please select an option (1-4): 4
Thank you for using the ATM. Goodbye!
```

Explanation:

This program simulates a simple ATM menu using loops and conditionals. It allows users to check their balance, deposit money, withdraw money, or exit the program. The program maintains an initial balance and updates it based on user actions while ensuring valid inputs are provided.