Experiment 4.2

Student Name: Anshu **UID: 22BCS16672**
Branch: BE-CSE **Section/Group: 642-A**
Semester:6th **Date of Performance:14/02/25**
Subject Name: PBLJ **Subject Code: 22CSH-359**

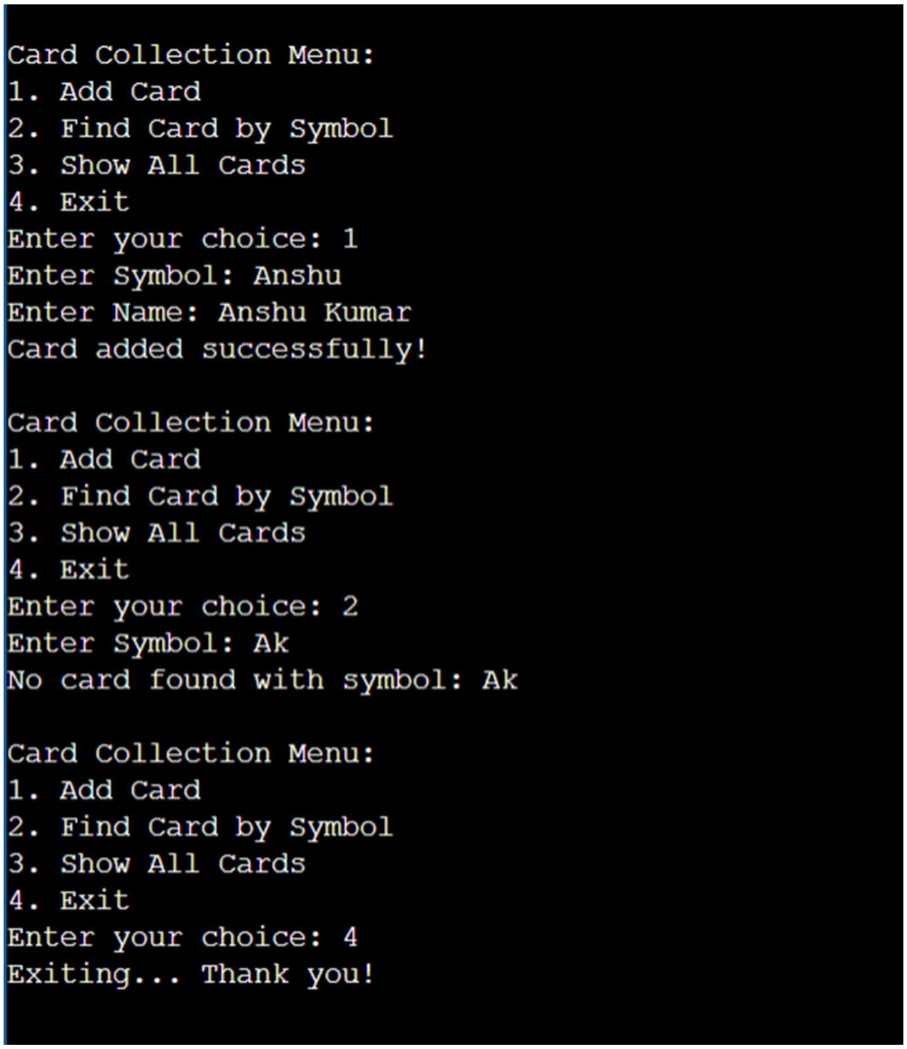
1. **Aim :** Write a program to collect and store all the cards to assist the users in finding all the cards in a given symbol.This cards game consist of N number of cards. Get N number of cards details from the user and store the values in Card object with the attributes symbol and Number. Store all the cards in a map with symbols as its key and list of cards as its value. Map is used here to easily group all the cards based on their symbol. Once all the details are captured print all the distinct symbols in alphabetical order from the Map.

2. **Objective :** This program collects and stores N cards, grouping them by symbol in a map for easy retrieval. It displays distinct symbols in alphabetical order along with their associated cards, total count, and sum of numbers, ensuring efficient organization and user-friendly output.

3. **Code**

```
import java.util.*;

class Card {
    String symbol;
    String name;

    // Constructor
    Card(String symbol, String name) {
        this.symbol = symbol;
        this.name = name;
    }

    // Override toString for better display
    public String toString() {
        return name + " (" + symbol + ")";
    }
}

public class CardCollection {
    static Collection<Card> cards = new ArrayList<>();
    static Scanner sc = new Scanner(System.in);

    public static void main(String[] args) {
        while (true) {
            // Display menu
            System.out.println("\nCard Collection Menu:");
            System.out.println("1. Add Card");
            System.out.println("2. Find Card by Symbol");
            System.out.println("3. Show All Cards");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");

            // Validate integer input
            while (!sc.hasNextInt()) {
                System.out.println("Invalid input! Please enter a number (1-4).");
                sc.next();
            }
            int choice = sc.nextInt();

            // Process user choice
            switch (choice) {
                case 1 -> addCard();
                case 2 -> findBySymbol();
                case 3 -> showAllCards();
                case 4 -> {
                    System.out.println("Exiting... Thank you!");
                    sc.close();
                    return;
                }
                default -> System.out.println("Invalid choice! Please select between
1-4.");
            }
        }
    }

    // Method to add a new card
    static void addCard() {
        System.out.print("Enter Symbol: ");
        String symbol = sc.next();
        sc.nextLine(); // Consume newline
        System.out.print("Enter Name: ");
        String name = sc.nextLine();
        cards.add(new Card(symbol, name));
        System.out.println("Card added successfully!");
    }

    // Method to find a card by symbol
    static void findBySymbol() {
        System.out.print("Enter Symbol: ");
        String symbol = sc.next();
        boolean found = false;
        for (Card c : cards) {
            if (c.symbol.equalsIgnoreCase(symbol)) { // Case-insensitive search
                System.out.println("Found: " + c);
                found = true;
            }
        }
        if (!found) {
            System.out.println("No card found with symbol: " + symbol);
        }
    }

    // Method to display all cards
    static void showAllCards() {
        if (cards.isEmpty()) {
            System.out.println("No cards in the collection.");
        } else {
            System.out.println("Cards in Collection:");
            cards.forEach(System.out::println);
        }
    }
}
```

4. Output



```
Card Collection Menu:
1. Add Card
2. Find Card by Symbol
3. Show All Cards
4. Exit
Enter your choice: 1
Enter Symbol: Anshu
Enter Name: Anshu Kumar
Card added successfully!

Card Collection Menu:
1. Add Card
2. Find Card by Symbol
3. Show All Cards
4. Exit
Enter your choice: 2
Enter Symbol: Ak
No card found with symbol: Ak

Card Collection Menu:
1. Add Card
2. Find Card by Symbol
3. Show All Cards
4. Exit
Enter your choice: 4
Exiting... Thank you!
```

5. Learning Outcomes

- Understand how to use maps (dictionaries) for efficient data storage and retrieval.
- Learn to group and organize data based on a key attribute.
- Gain experience in handling user input and storing objects dynamically.