Experiment 4

Student Name: Amir Raza **UID: 22BCS12331**
Branch: BE-CSE **Section/Group: 639-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

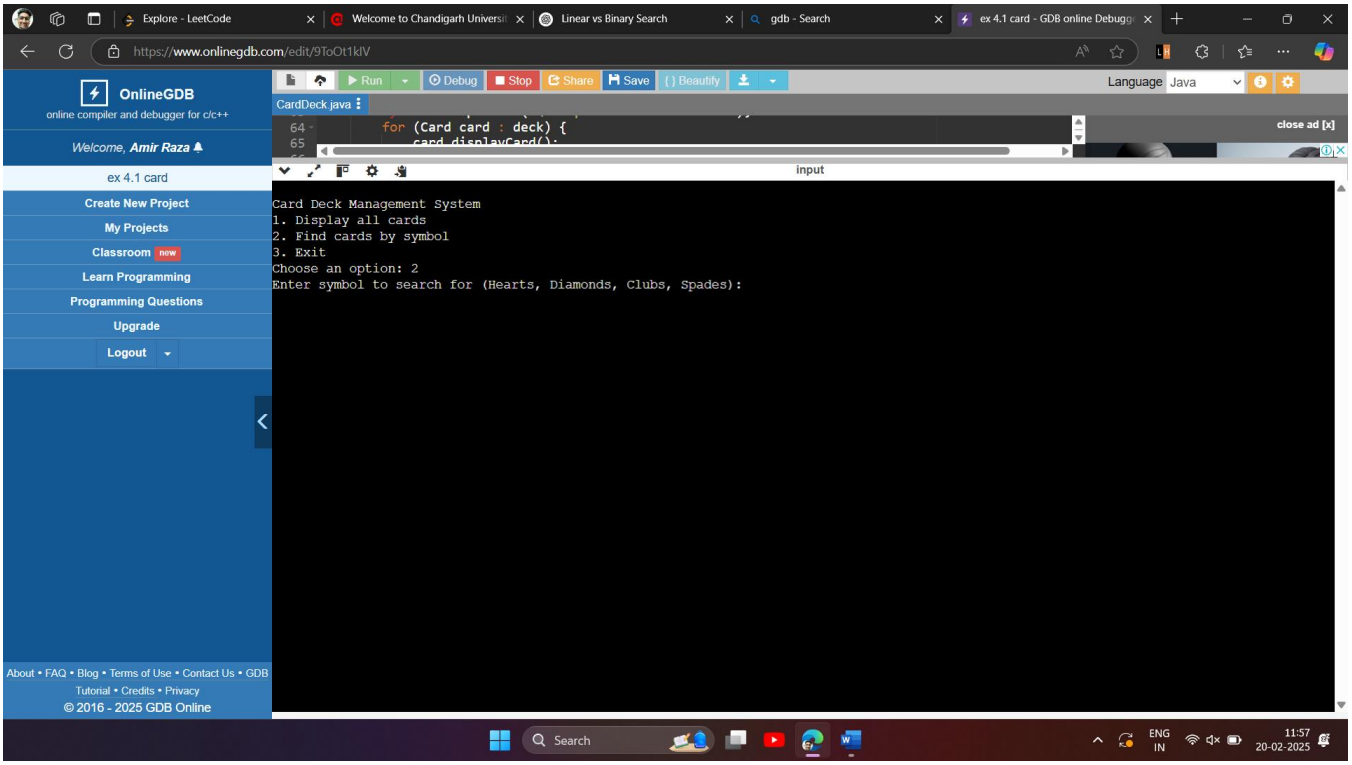
```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Card {
    String symbol;
    String rank;

    // Constructor to initialize the card
    public Card(String symbol, String rank) {
        this.symbol = symbol;
        this.rank = rank;
    }

    // Method to display card information
    public void displayCard() {
        System.out.println("Card: " + rank + " of " + symbol);
    }

    // Getter for symbol
    public String getSymbol() {
```

```
        return symbol;
    }
}

public class CardDeck {
    private List<Card> deck;

    // Constructor to initialize deck of cards
    public CardDeck() {
        deck = new ArrayList<>();
        createDeck();
    }

    // Create a deck with 4 suits and 13 ranks
    private void createDeck() {
        String[] symbols = {"Hearts", "Diamonds", "Clubs", "Spades"};
        String[] ranks = {"2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen",
"King", "Ace"};

        for (String symbol : symbols) {
            for (String rank : ranks) {
                deck.add(new Card(symbol, rank));
            }
        }
    }

    // Method to find and display cards with a given symbol
    public void findCardsBySymbol(String symbol) {
        boolean found = false;
        for (Card card : deck) {
            if (card.getSymbol().equalsIgnoreCase(symbol)) {
                card.displayCard();
                found = true;
            }
        }
        if (!found) {
            System.out.println("No cards found with the symbol: " + symbol);
        }
    }

    // Method to display all cards in the deck
    public void displayAllCards() {
```

```
        System.out.println("\nComplete deck of cards:");
        for (Card card : deck) {
            card.displayCard();
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        CardDeck cardDeck = new CardDeck();

        int choice;
        do {
            System.out.println("\nCard Deck Management System");
            System.out.println("1. Display all cards");
            System.out.println("2. Find cards by symbol");
            System.out.println("3. Exit");
            System.out.print("Choose an option: ");
            choice = scanner.nextInt();
            scanner.nextLine();  // Consume newline

            switch (choice) {
                case 1:
                    cardDeck.displayAllCards();
                    break;

                case 2:
                    System.out.print("Enter symbol to search for (Hearts, Diamonds, Clubs,
Spades): ");
                    String symbol = scanner.nextLine();
                    cardDeck.findCardsBySymbol(symbol);
                    break;

                case 3:
                    System.out.println("Exiting the program...");
                    break;

                default:
                    System.out.println("Invalid option. Please try again.");
            }

        } while (choice != 3);
```

```
        scanner.close();
    }
}
```

4.Code output



1. 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.

- Develop skills in sorting and displaying structured data in a meaningful