



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment No: 2.1

**Student Name:** Sachin Kumar Singh

**UID:** 21BCS9217

**Branch:** BE-CSE

**Section/Group:** 21BCS-CC-646 (B)

**Semester:** 6<sup>th</sup>

**Date of Performance:** 05/02/2024

**Subject Name:** Project Based Learning in Java with Lab

**Subject Code:** 21CSH-319

**Aim:** Write a program to collect and store all the cards to assist the users in finding all the cards in a given

### Objective:

1. To learn about Classes.
2. To learn about hashing in java.

**Input/Apparatus Used:** IntelliJ / VS Code.

### Procedure/Algorithm/Pseudocode:

#### 1. Card Class:

Represents a card with a symbol (e.g., ♠, ♦, ♣, ♥) and a numerical value.  
Provides methods to get and set the card symbol and value.

#### 2. PlayGame Class:

Manages the gameplay, including creating cards, calculating totals, and displaying results.  
Contains an ArrayList to store Card objects and a HashMap to keep track of the total value of cards for each symbol.

#### 3. createCard Method:

Adds a new card with the given symbol and value to the list of cards.



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## **4. calculateTotal Method:**

Iterates through the list of cards.

Calculates the total value of cards for each symbol and stores it in the HashMap.

## **5. display Method:**

Calls calculateTotal to ensure totals are up to date.

Displays the distinct symbols present in the card set.

For each symbol, displays the cards present, their values, the number of cards, and the sum of their values.

## **6. play Method:**

Asks the user for the number of cards they want to input.

For each card, prompts the user to choose a symbol and value.

Validates the input values and adds the card to the game if they are valid.

## **7. Main Class:**

Contains the main method where the game is initiated.

Creates an instance of the PlayGame class, takes user input for card details, and displays the results



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Code:

```
package University.Java_Using_Project.Exprtiment4;
```

```
import java.util.*;
```

```
class Card {
```

```
    private char cardSymbol;  
    private int value;
```

```
    public Card(char cardSymbol, int value) {  
        this.cardSymbol = cardSymbol;  
        this.value = value;  
    }
```

```
    public char getCardSymbol() {  
        return cardSymbol;  
    }
```

```
    public void setCardSymbol(char cardSymbol) {  
        this.cardSymbol = cardSymbol;  
    }
```

```
    public int getValue() {  
        return value;  
    }
```

```
    public void setValue(int value) {  
        this.value = value;  
    }  
}
```

```
class PlayGame {
```

```
    ArrayList<Card> cards = new ArrayList<>();
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
Map<Character, Integer> map = new HashMap<>();

void createCard(char symbol, int value) {
    cards.add(new Card(symbol, value));
}

void calculateTotal() {

    for (Card c : cards) {
        if (c.getCardSymbol() == '♠') {
            int cardValue = c.getValue();
            int value = map.getDefault('♠', 0);
            map.put('♠', value + cardValue);
        } else if (c.getCardSymbol() == '♦') {
            int cardValue = c.getValue();
            int value = map.getDefault('♦', 0);
            map.put('♦', value + cardValue);
        } else if (c.getCardSymbol() == '♣') {
            int cardValue = c.getValue();
            int value = map.getDefault('♣', 0);
            map.put('♣', value + cardValue);
        } else if (c.getCardSymbol() == '♥') {
            int cardVale = c.getValue();
            int value = map.getDefault('♥', 0);
            map.put('♥', value + cardVale);
        }
    }
}

void display() {
    calculateTotal();
    System.out.println("\nDistinct Symbols are:");
    Set<Character> keys = map.keySet();
    for (Character key : keys) {
        System.out.print(key + " ");
    }
    System.out.println();
    for (Map.Entry<Character, Integer> e : map.entrySet()) {
        System.out.println("Cards in " + e.getKey() + " Symbol");
    }
}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        int cardCount = 0;
        for (Card c : cards) {
            if (c.getCardSymbol() == e.getKey()) {
                System.out.println(c.getCardSymbol() + " : " + c.getValue());
                cardCount++;
            }
        }
        System.out.println("Number of " + e.getKey() + " cards: " + cardCount);
        System.out.println("Sum of Number: " + e.getValue() + "\n");
    }
}

void play(Scanner in) {
    System.out.println("Enter the number of cards");
    int numberOfCards = in.nextInt();
    int i = 0;
    while (i < numberOfCards) {

        System.out.println("\nEnter 1 for SPADE");
        System.out.println("Enter 2 for DIAMOND");
        System.out.println("Enter 3 for CLUB");
        System.out.println("Enter 4 for HEART\n");
        if (i == 0) {
            System.out.print("Enter 1st card: ");
        } else if (i == 1) {
            System.out.print("Enter 2nd card: ");
        } else if (i == 2) {
            System.out.print("Enter 3rd card: ");
        } else {
            System.out.print("\nEnter " + (i + 1) + "th card: ");
        }
        int choice = in.nextInt();
        switch (choice) {
            case 1: {
                char symbol = '♠';
                System.out.print("Enter the value of SPADE card: ");
                int value = in.nextInt();
                if (1 <= value && value <= 13) {
                    createCard(symbol, value);
                    i++;
                } else {
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        System.out.println("Value is invalid!!!");
    }

}
break;
case 2: {
    char symbol = '♦';
    System.out.print("Enter the value of DIAMOND card: ");
    int value = in.nextInt();
    if (1 <= value && value <= 13) {
        createCard(symbol, value);
        i++;
    } else {
        System.out.println("Value is invalid!!!");
    }
}
break;
case 3: {
    char symbol = '♣';
    System.out.print("Enter the value of CLUB card: ");
    int value = in.nextInt();
    if (1 <= value && value <= 13) {
        createCard(symbol, value);
        i++;
    } else {
        System.out.println("Value is invalid!!!");
    }
}
break;
case 4: {
    char symbol = '♥';
    System.out.print("Enter the value of HEART card: ");
    int value = in.nextInt();
    if (1 <= value && value <= 13) {
        createCard(symbol, value);
        i++;
    } else {
        System.out.println("Value is invalid!!!\n");
    }
}
default: {
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        System.out.println("WRONG choice");
    }
    break;
}
}
}
```

```
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        PlayGame game = new PlayGame();
        game.play(in);
        game.display();
    }
}
```

## Result/Output:

Distinct Symbols are:

♠ ♣ ♥ ♦

Cards in ♠ Symbol

♠ : 1

♠ : 12

♠ : 13

♠ : 7

Number of ♠ cards: 4

Sum of Number: 33

Cards in ♣ Symbol

♣ : 5

♣ : 3

♣ : 2

Number of ♣ cards: 3

Sum of Number: 10

Cards in ♥ Symbol

♥ : 5

♥ : 7

♥ : 9

Number of ♥ cards: 3

Sum of Number: 21

Cards in ♦ Symbol

♦ : 4

♦ : 4

♦ : 3

Number of ♦ cards: 3

Sum of Number: 11