DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Experiment 2.2

Student Name: Jitesh Kumar **UID: 20BCS2334**
Branch: CSE **Section/Group: 20BCS_WM-903_A**
Semester: 5 **Date of Performance: 28/9/2022**
Subject Name: PBLJ Lab **Subject Code: 20CSP-321**

1. Aim/Overview of the practical:

Collect Unique Symbols From Set of Cards

Playing cards during travel is a fun filled experience. For this game they wanted to collect all four unique symbols. Can you help these guys to collect unique symbols from a set of cards?

Create Card class with attributes symbol and number. From our main method collect each card details (symbol and number) from the user.

Collect all these cards in a set, since set is used to store unique values or objects.

Once we collect all four different symbols display the first occurrence of card details in alphabetical order.

2. Software/Hardware Requirements:

- Laptop
- Eclipse IDE

3. Algorithm/pseudo code:

Step 1- Start

Step 2- Create a class Card which implements Comparable interface

Step 3- Class contains symbol and number data members, getter and setter functions.

Step 4- Declare a HashSet which take values of type Card.

Step 5- Take 8 input of symbols and numbers from user and create objects and using constructor initialize their values and add object in Set.

Step 6- Display the first occurrences of four cards values in the set

Step 7- Exit

4.Steps for experiment/practical/Code:

Card.java

```
package com;
public class Card implements Comparable<Card> {
	private char symbol;
	private int number;

	public Card() {}

	public Card(char symbol, int number) {
		super();
		this.symbol = symbol;
		this.number = number;
	}

	public char getSymbol() {
```

```
        return symbol;
}

public void setSymbol(char symbol) {
        this.symbol = symbol;
}

public int getNumber() {
        return number;
}

public void setNumber(int number) {
        this.number = number;
}

@Override
public String toString() {
        return "Card [symbol=" + symbol + ", number=" + number + "]";
}

@Override
public int compareTo(Card o) {
        if (this.symbol < o.symbol) return -1;
```

```
            else if (this.symbol > o.symbol) return 1;
            else return 1;
        }

        @Override
        public int hashCode() {
       return String.valueOf(symbol).hashCode();
        }

        @Override
    public boolean equals(Object obj){
        if (obj instanceof Card) {
          Card card = (Card) obj;
           return (card.symbol == this.symbol);
        } else {
           return false;
        }
    }
}
```

Main.java

```
package com;
import java.util.*;
import java.lang.*;
import java.io.*;
```

```
import com.Card;
public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Set<Card> set = new HashSet<>();

        for (int i = 0; i < 8; i++) {
            System.out.println("Enter a card:");
            Card card = new Card();

            card.setSymbol(sc.nextLine().charAt(0));
            card.setNumber(sc.nextInt());
            sc.nextLine();

            set.add(card);
        }
        System.out.println("Four symbols gathered in
eight cards.");
        System.out.println("Cards in Set are:");


        for (Card card : set)
            System.out.println(card.getSymbol() + " "
+ card.getNumber());

        sc.close();
    }

}
```

5. Result/Output/Writing Summary:

```
Enter a card:
A
1
Enter a card:
A
2
Enter a card:
A
7
Enter a card:
D
6
Enter a card:
C
2
Enter a card:
D
1
Enter a card:
C
1
Enter a card:
B
2
Four symbols gathered in eight cards.
Cards in Set are:
A 1
B 2
C 2
D 6
```

6.Learning outcomes (What I have learnt):

1. **Learnt about use of collections, HashSet**
2. **Learnt about various operation of HashSet.**
3. **Learnt about Comparable interface**