**Department of Computer Science & Engineering**



# UNIVERSITY INSTITUTE OF ENGINEERING

## Department of Computer Science & Engineering

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

**Subject Code:** 20CSP321

**Submitted to: Er.Parveen Tanwar Sir**                     **Submitted by: Priya Bharti**

**Faculty name**: Er. Parveen Tanwar Sir                     **Name**: Priya Bharti

                                                             **UID:** 20BCS3524

                                                             **Section:** 607

                                                             **Group:** B

**Department of Computer Science & Engineering**

# Department of Computer Science & Engineering
## INDEX

| Ex. No | List of Experiments | Conduct (MM: 12) | Viva (MM: 10) | Record (MM: 8) | Total (MM: 30) | Date | Remarks/Signature |
|---|---|---|---|---|---|---|---|
| 1.1 | Create an application to save the employee information using arrays. | | | | | 03/09/22 | |
| 1.2 | Design and implement a simple inventory control system for a small video rental store. | | | | | 05/09/22 | |
| 1.3 | Create a application to calculate interest for FDs, RDs based on certain conditions using inheritance. | | | | | 10/09/22 | |
| 2.1 | Create a program to set view of Keys from Java Hashtable. | | | | | 29/09/22 | |
| 2.2 | Create a program to show the usage of Sets of Collection interface. | | | | | 07/10/22 | |
| 2.3 | Write a Program to perform them basic operations like insert, delete, display, and search in list. List contains String object items where these operations are to be performed. | | | | | 12/10/22 | |
| 2.4 | Create a menu-based Java application with the following options. 1.Add an Employee 2.Display All 3.Exit If option 1 is selected, the application should gather details of the employee like employee name, employee id, designation and salary and store it in a file. If option 2 is selected, the application should display all the employee details. If option 3 is selected the application should exit. | | | | | 13/10/22 | |
| 3.1 | Create a palindrome creator application for making a longest possible palindrome out of given input string. | | | | | | |
| 3.2 | Create a Servlet/ application with a facility to print any message on web browser. | | | | | | |
| 3.3 | Create JSP application for addition, multiplication and division. | | | | | | |

## Department of Computer Science & Engineering

## Experiment 2.1

**Student Name: Priya Bharti**          UID: 20BS3524

**Branch: CSE**          **Section/Group: 607-B**

**Semester: 5th**          **Date of Performance: 29/09/22**

**Subject Name: PBLJ Lab**          **Subject Code: 20CSP-321**

**AIM:**

Write a program to collect and store all the cards to assist the users in finding all the cards in a given symbol.

**OBJECTIVE:**

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 symbol 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. For each symbol print all the card details, number of cards and their sum respectively.

Sample input output :

Enter Number of Cards :

13

Enter card 1:

s

1

Enter card 2:

s

12

Enter card 3:

s

13

Enter card 4:

d

4

Enter card 5:

c

5

Enter card 6:

h

5

Enter card 7:

h

7

Enter card 8:

c

3

Enter card 9:

c

2

Enter card 10:

h

9

Enter card 11:

s

7

Enter card 12:

d

4

Enter card 13:

d

3

Distinct Symbols are :

c d h s

Cards in c Symbol

c 5

c 3

c 2

Number of cards : 3

Sum of Numbers : 10

Cards in d Symbol

d 4

d 4

d 3

Number of cards : 3

Sum of Numbers : 11

Cards in h Symbol

h 5

h 7

h 9

Number of cards : 3

Sum of Numbers : 21

Cards in s Symbol

s 1

s 12

s 13

s 7

Number of cards : 4

Sum of Numbers : 33

**JAVA CODE/INPUT:**

# TestMain.java

package com.w3epic.service;

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Scanner;
import java.util.Set;
import java.util.TreeMap;

import com.w3epic.bean.Card;

public class TestMain {

    public static void main(String[] args) {
```

```java
Scanner sc = new Scanner(System.in);
Map<Character, ArrayList<Card>> map = new TreeMap<>();

System.out.println("Enter Number of Cards :");
int n = sc.nextInt();
sc.nextLine();

for (int i = 1; i <= n; i++) {
    System.out.println("Enter card " + i);
    char symbol = sc.nextLine().charAt(0);
    int number = sc.nextInt();

    Card card = new Card();
    card.setSymbol(symbol);
    card.setNumber(number);
    sc.nextLine();

    if (!map.containsKey(symbol)) {
        ArrayList<Card> list = new ArrayList<>();
        list.add(card);
        map.put(symbol, list);
    } else {
        ArrayList<Card> list = map.get(symbol);
        list.add(card);
    }
}
System.out.println("Distinct Symbols are :");

Set<Entry<Character, ArrayList<Card>>> set = map.entrySet();
Iterator<Entry<Character, ArrayList<Card>>> it = set.iterator();
while (it.hasNext()) {
    System.out.print(it.next().getKey() + " ");
}
System.out.println();


set = map.entrySet();
it = set.iterator();

while (it.hasNext()) {
    int sum = 0;
    Map.Entry<Character, ArrayList<Card>> me = it.next();
    ArrayList<Card> list = me.getValue();

    System.out.println("Cards in " + me.getKey() + " Symbol");

    for (Card card : list) {
        System.out.println(card.getSymbol() + " " + card.getNumber());
        sum += card.getNumber();
```

```
        }

        System.out.println("Number of cards : " + list.size());
        System.out.println("Sum of Numbers : " + sum);
    }

    sc.close();
  }

}
```

## Card.java

```java
package com.w3epic.bean;

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 + "]";
    }
```
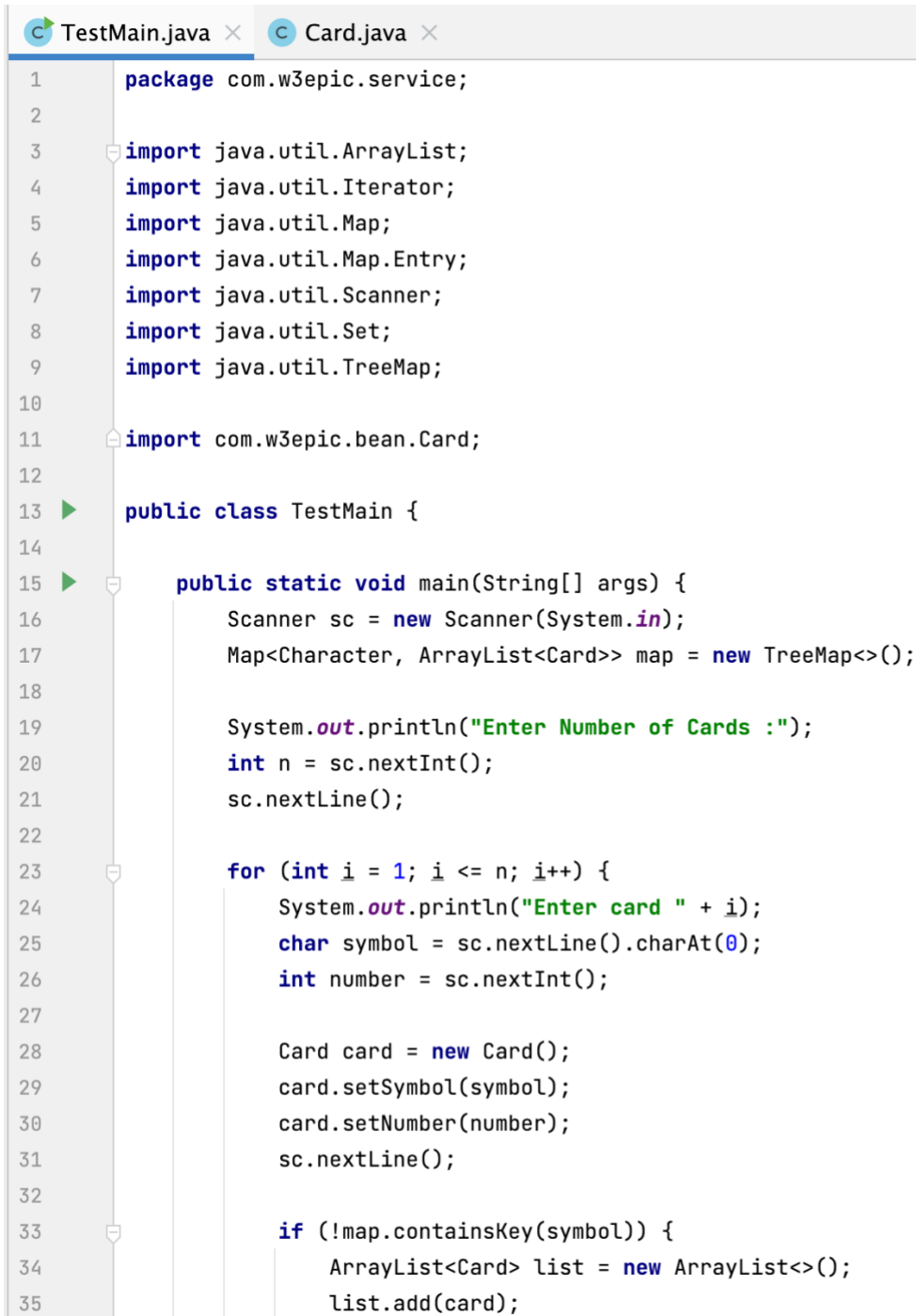
```java
@Override
public int compareTo(Card o) {
    if (this.symbol < o.symbol) return -1;
    else if (this.symbol > o.symbol) return 1;
    else return 1;
}
```

TestMain.java ×    C Card.java ×

```java
1        package com.w3epic.service;
2
3        import java.util.ArrayList;
4        import java.util.Iterator;
5        import java.util.Map;
6        import java.util.Map.Entry;
7        import java.util.Scanner;
8        import java.util.Set;
9        import java.util.TreeMap;
10
11       import com.w3epic.bean.Card;
12
13       public class TestMain {
14
15           public static void main(String[] args) {
16               Scanner sc = new Scanner(System.in);
17               Map<Character, ArrayList<Card>> map = new TreeMap<>();
18
19               System.out.println("Enter Number of Cards :");
20               int n = sc.nextInt();
21               sc.nextLine();
22
23               for (int i = 1; i <= n; i++) {
24                   System.out.println("Enter card " + i);
25                   char symbol = sc.nextLine().charAt(0);
26                   int number = sc.nextInt();
27
28                   Card card = new Card();
29                   card.setSymbol(symbol);
30                   card.setNumber(number);
31                   sc.nextLine();
32
33                   if (!map.containsKey(symbol)) {
34                       ArrayList<Card> list = new ArrayList<>();
35                       list.add(card);
```

```java
36                    map.put(symbol, list);
37              } else {
38                  ArrayList<Card> list = map.get(symbol);
39                  list.add(card);
40              }
41          }
42          System.out.println("Distinct Symbols are :");
43
44          Set<Entry<Character, ArrayList<Card>>> set = map.entrySet();
45          Iterator<Entry<Character, ArrayList<Card>>> it = set.iterator();
46          while (it.hasNext()) {
47              System.out.print(it.next().getKey() + " ");
48          }
49          System.out.println();
50
51          set = map.entrySet();
52          it = set.iterator();
53
54          while (it.hasNext()) {
55              int sum = 0;
56              Map.Entry<Character, ArrayList<Card>> me = it.next();
57              ArrayList<Card> list = me.getValue();
58              System.out.println("Cards in " + me.getKey() + " Symbol");
59
60              for (Card card : list) {
61                  System.out.println(card.getSymbol() + " " + card.getNumber());
62                  sum += card.getNumber();
63              }
64
65              System.out.println("Number of cards : " + list.size());
66              System.out.println("Sum of Numbers : " + sum);
67          }
68          sc.close();
69      }
70  }
```

```java
package com.w3epic.bean;

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;
    }
}
```

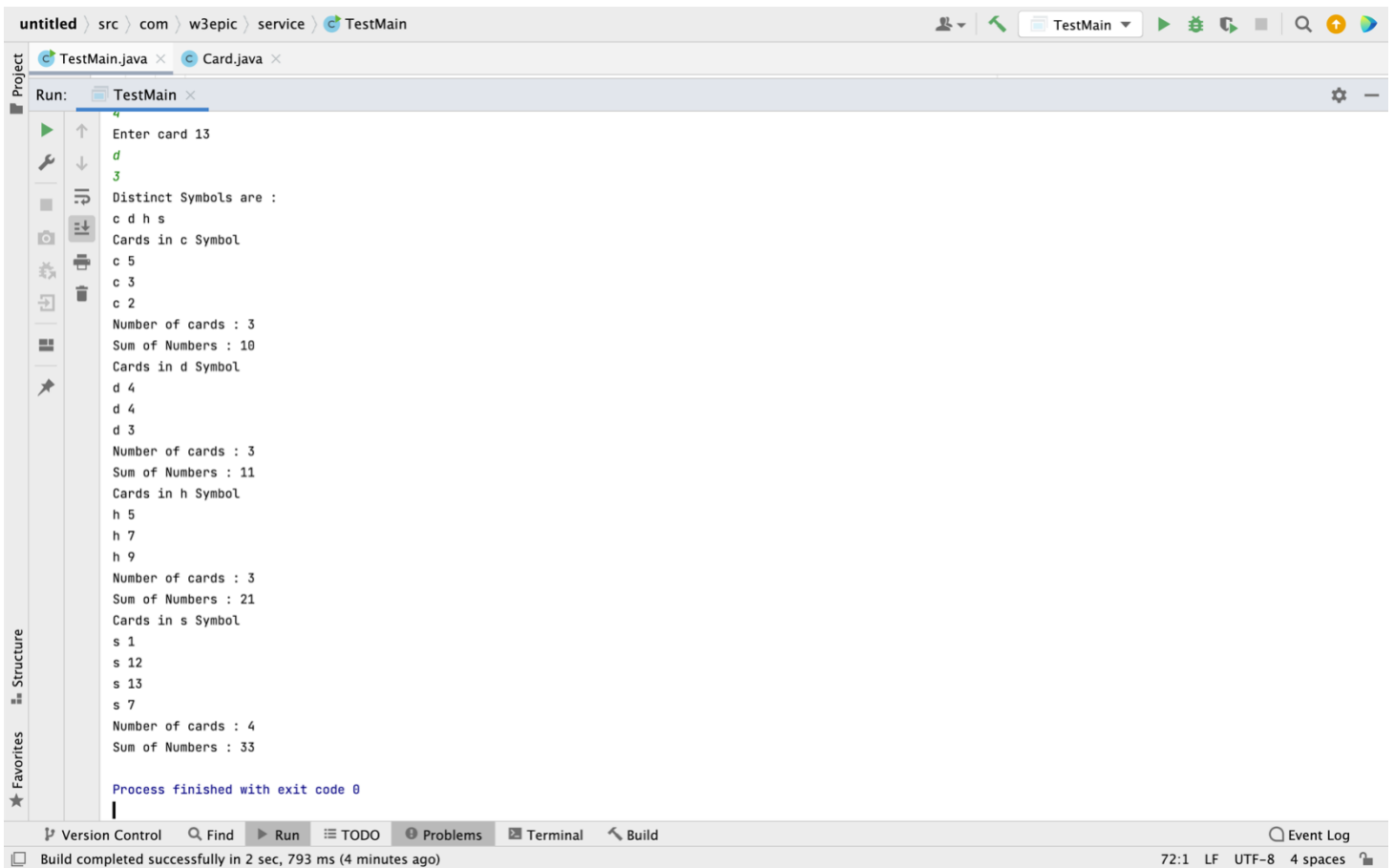# Department of Computer Science & Engineering

**OUTPUT:**



```
untitled ) src ) com ) w3epic ) service ) TestMain

TestMain.java ×    Card.java ×

Run:    TestMain ×

/Library/Java/JavaVirtualMachines/jdk1.8.0_301.jdk/Contents/Home/bin/java ...
Enter Number of Cards :
13
Enter card 1
s
1
Enter card 2
s
12
Enter card 3
s
13
Enter card 4
d
4
Enter card 5
c
5
Enter card 6
h
5
Enter card 7
h
7
Enter card 8
c
3
Enter card 9
c
2
Enter card 10
h
9
Enter card 11
```

Build completed successfully in 2 sec, 793 ms (2 minutes ago)    72:1  LF  UTF-8  4 spaces

# Department of Computer Science & Engineering

TestMain.java    Card.java

Run: TestMain

```
Enter card 11
s
7
Enter card 12
d
4
Enter card 13
d
3
Distinct Symbols are :
c d h s
Cards in c Symbol
c 5
c 3
c 2
Number of cards : 3
Sum of Numbers : 10
Cards in d Symbol
d 4
d 4
d 3
Number of cards : 3
Sum of Numbers : 11
Cards in h Symbol
h 5
h 7
h 9
Number of cards : 3
Sum of Numbers : 21
Cards in s Symbol
s 1
s 12
s 13
```

Version Control    Find    Run    TODO    Problems    Terminal    Build    Event Log

Build completed successfully in 2 sec, 793 ms (3 minutes ago)    72:1    LF    UTF-8    4 spaces

```
untitled  src  com  w3epic  service   TestMain                                    TestMain

 TestMain.java    Card.java

Run:      TestMain

     Enter card 13
     d
     3
     Distinct Symbols are :
     c d h s
     Cards in c Symbol
     c 5
     c 3
     c 2
     Number of cards : 3
     Sum of Numbers : 10
     Cards in d Symbol
     d 4
     d 4
     d 3
     Number of cards : 3
     Sum of Numbers : 11
     Cards in h Symbol
     h 5
     h 7
     h 9
     Number of cards : 3
     Sum of Numbers : 21
     Cards in s Symbol
     s 1
     s 12
     s 13
     s 7
     Number of cards : 4
     Sum of Numbers : 33

     Process finished with exit code 0

 Version Control   Find   Run   TODO   Problems   Terminal   Build                Event Log
 Build completed successfully in 2 sec, 793 ms (4 minutes ago)              72:1  LF  UTF-8  4 spaces
```

## Learning outcomes (What I have learnt):

1. To write a program in java to store and collect all the card.
2. Concept of arraylist.
3. Loops concept.
4. Map, Set, and TreeMap concepts.