Lab15-Assignment

Name : M Bhagyalaxmi Student id:AF0339449

1)Write a Java program that demonstrates the following operations on a HashSet: • Create a HashSet of integers. • Add the numbers 5, 10, 15, 20, and 25 to the set. • Display the elements of the set. • Check if the set contains the number 10. • Remove the number 15 from the set. • Display the size of the set.

```
package com.wrapperclass.examples;
import java.util.HashSet;
import java.util.Scanner;
public class Hashset {
public static void main(String[] args) {

          HashSet<Integer> hs = new HashSet<Integer>();
          hs.add(5);
          hs.add(10);
          hs.add(15);
          hs.add(20);
          hs.add(25);

          //display
          System.out.println(hs);

          //contains() usage
          System.out.println("Collection contains Element
10 :"+ hs.contains(10));

          //remove an element
          hs.remove(15);

          //display size of collection
          System.out.println("Size of collection "+
hs.size());


     }
```

}

Output:

```
1  package com.wrapperclass.examples;
2  import java.util.HashSet;
3  import java.util.Scanner;
4  public class Hashset {
5  public static void main(String[] args) {
6
7          HashSet<Integer> hs = new HashSet<Integer>();
8          hs.add(5);
9          hs.add(10);
10         hs.add(15);
11         hs.add(20);
12         hs.add(25);
13
14         //display
15         System.out.println(hs);
16
17         //contains() usage
18         System.out.println("Collection contains Element 10 :"+ hs.contains(10));
19
```

Problems Javadoc Declaration Console Coverage

<terminated> Hashset [Java Application] C:\Users\USER\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_19.0.2.v20230129-1123\jre\bin\javaw.exe (Nov 16, 2023,

```
[20, 5, 25, 10, 15]
Collection contains Element 10 :true
Size of collection 4
```

2) Write a Java program to simulate the "Hot Potato" game using a queue. In this game, players stand in a circle and pass a potato (or any other object) while music plays. When the music stops, the player holding the potato is out. ● Create a queue to represent the circle of players. ● Enqueue player names. ● Dequeue players one by one and enqueue them again to simulate passing the potato. ● Repeat the dequeue and enqueue process, simulating the music stopping and players being eliminated until only one player remains.

```
package com.wrapperclass.examples;
import java.util.LinkedList;
import java.util.Queue;
public class HotPotatoGame {
      public static void main(String[] args) {
            // Create a queue to represent the circle
of players
            Queue<String> playersQueue = new
LinkedList<>();

            // Enqueue player names
```

```
            playersQueue.add("Bhagya");
            playersQueue.add("vasantha");
            playersQueue.add("krishna");
            playersQueue.add("venu");
            playersQueue.add("vijay");

            // Set the number of rounds to simulate the
game
            int rounds = 3;

            simulateHotPotatoGame(playersQueue,
rounds);
        }

        private static void
simulateHotPotatoGame(Queue<String> playersQueue, int
rounds) {
            System.out.println("Starting Hot Potato
Game!");

            while (playersQueue.size() > 1) {
                System.out.println("Current players in
the circle: " + playersQueue);

                // Simulate passing the potato by
dequeueing and enqueuing
                for (int i = 0; i < rounds; i++) {
                    String currentPlayer =
playersQueue.poll();
                    System.out.println("Passing the
potato from " + currentPlayer + " to the next
player.");
                    playersQueue.offer(currentPlayer);
                }

                // The player holding the potato is
eliminated
                String eliminatedPlayer =
playersQueue.poll();
```

```
                System.out.println(eliminatedPlayer + "
is out!");

                System.out.println();
            }

            // The last remaining player wins the game
            System.out.println("The winner is: " +
playersQueue.poll());
        }

}
```

Output: