**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**.

// program for the operation of hashset

package colleection;

import java.util.HashSet;

public class HashSetOperation

{

public static void main(String[] args)

{

HashSet<Integer> numbers = new HashSet<>();

//adding numbers to hashset

numbers.add(5);

numbers.add(10);

numbers.add(15);

numbers.add(20);

numbers.add(25);

// printing the added integers of hashset

System.out.println("Elements in the HashSet: " + numbers);

// check whether 10 preset

System.out.println("Does the set contain 10? " + numbers.contains(10));

// 15 remove from hashset

numbers.remove(15);

// printing size of hashset

System.out.println("Size of the HashSet: " + numbers.size());

}

}

**Output:**

Elements in the HashSet: [20, 5, 25, 10, 15]

Does the set contain 10? true

Size of the HashSet: 4

**2. Write a Java program to simulate the "Hot Potato" game using a queue. In this game.**

// program for potato game using queue and perform enqueue and dequeue for win

package colleection;

import java.util.LinkedList;

import java.util.Queue;

import java.util.Random;

public class HotGames

{

public static void main(String[] args)

{

// crating queue and adding elements

Queue<String> players = new LinkedList<>();

players.add("Player1");

players.add("Player2");

players.add("Player3");

players.add("Player4");

players.add("Player5");

players.add("Player6");

// choose random for random player win

Random random = new Random();

// running loop to eliminate all except 1

while (players.size() > 1)

{

int count = random.nextInt(players.size());

// adding randomly and removing

for (int i = 0; i < count; i++)

{

String currentPlayer = players.poll();

players.add(currentPlayer);

}

String eliminatedPlayer = players.poll();

System.out.println("Player eliminated: " + eliminatedPlayer);

}

// printing winner

System.out.println("Winner: " + players.poll());

}

}

**OUTPUT:**

**1.**

Player eliminated: Player1

Player eliminated: Player2

Player eliminated: Player3

Player eliminated: Player5

Winner: Player6

**2.**

Player eliminated: Player5

Player eliminated: Player2

Player eliminated: Player4

Player eliminated: Player3

Player eliminated:Player6

Winner: Player1