Experiment:10

Aim: Write a Java Program to implement Simple Queue to
1)To add Items into the queue.
2)To delete items from the queue.

Program:

```
import java.util.*;
// define queue class
class Queue
{
   int arr[], front, rear, cap, n1;
          // Queue constructor
        Queue(int n)
        {
                arr = new int[n];
                cap = n;
                front = 0;
                rear = -1;
                n = 0;
        }
        // dequeue function for removing the front element
        public void dequeue()
        {
                // check for queue underflow
                if (isEmpty())
                {
                        System.out.println("No items in the queue,cannot delete");
                        System.exit(1);
                }
                System.out.println("Deleted Element is " + arr[front]);
                front = (front + 1) % cap;
                n1--;
        }
```

```
    // enqueue function for adding an item to the rear
public void enqueue(int val)
    {
        // check for queue overflow
        if (isFull())
        {
            System.out.println("OverFlow!!Cannot add more values");
            System.exit(1);
        }

        System.out.println("Element Inserted " + val);
        rear = (rear + 1) % cap;
        arr[rear] = val;
        n1++;
    }
    // peek function to return front element of the queue
    public int peek()
    {
        if (isEmpty())
        {
            System.out.println("Queue empty!!Cannot delete");
            System.exit(1);
        }
        return arr[front];
    }
    // returns the size of the queue
    public int size()
    {
        return n1;
    }

    // to check if the queue is empty or not
    public Boolean isEmpty()
```

```
        {
                return (size() == 0);
        }
        // to check if the queue is full or not
        public Boolean isFull()
        {return (size() == cap);}
}
        // Queue implementation in java
  class Queuedemo{
        public static void main (String[] args)
        {
                // create a queue of capacity 5
                Queue q = new Queue(5);
                q.enqueue(10);
                q.enqueue(20);
                q.enqueue(30);
                System.out.println("Front element is: " + q.peek());
                q.dequeue();
                System.out.println("Front element is: " + q.peek());
                System.out.println("Queue size is " + q.size());
                q.dequeue();
                q.dequeue();

                if (q.isEmpty())
                        System.out.println("Queue Is Empty");
                else
                        System.out.println("Queue Is Not Empty");
        }
}
```