



//By Ansuman Swain

CSE 2001: Data Structure & Algorithms  
Programming Assignment-VII  
(Queue)

// you can visit github by this link and check out all codes  
// from Assignment 1 to Assignment 7

[https://github.com/Ansul2chiku/DSA\\_Sem2\\_Assignments](https://github.com/Ansul2chiku/DSA_Sem2_Assignments) <- *Click on this link  
to get code files...*



## Question-1

```
package Assignment_7;
import java.util.*;
public class QueueDemo1<T>
{
    public static final int MAX = 5;
    public static int front = -1;
    public static int rear = -1;
    public static <T> void insert(T Q[])
    {
        if (is_full())
        {
            System.out.println("Queue is full, Cannot insert element.");
            return;
        }
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter the element to insert:");
        T element = (T) obj.next();
        if (is_empty())
            front = 0;
        Q[++rear] = element;
        System.out.println("Element " + element + " inserted successfully.");
    }
    public static <T> void delete(T Q[])
    {
        if (is_empty())
        {
            System.out.println("Queue is empty, Cannot delete element...");
            return;
        }
        System.out.println("Element " + Q[front] + " deleted successfully.");
        if (rear == 0)
            front = rear=-1;
        else
        {
            for (int i = 1; i <=rear; i++)
                Q[i-1] = Q[i];
            rear--;
        }
    }
}
```



## Question-1

```
public static <T> void display(T Q[])
{
    if (is_empty())
    {
        System.out.println("Queue is empty...Nothing to display");
        return;
    }
    System.out.println("Elements in the queue:");
    for (int i = front; i <= rear; i++)
        System.out.print(Q[i] + " <- ");
    System.out.println();
}
public static boolean is_full()
{
    return rear == MAX - 1;
}
public static boolean is_empty()
{
    return front == -1;
}
public static void main(String[] args) {
    Scanner obj = new Scanner(System.in);
    Object queue[] = new Object[MAX];
    while (true)
    {
        System.out.println("****MENU****");
        System.out.println("0: Exit");
        System.out.println("1: Insert");
        System.out.println("2: Delete");
        System.out.println("3: Display");
        System.out.println("Enter your choice:");
        int choice = obj.nextInt();
        switch (choice)
        {
            case 0:
                System.exit(0);
            case 1:
                insert(queue);
                break;
            case 2:
                delete(queue);
                break;
            case 3:
                display(queue);
                break;
            default:
                System.out.println("Invalid choice");
        }
    }
}
```



## Question-2

```
package Assignment_7;
import java.util.*;
class Node<T>
{
    T info;
    Node<T> next;
    Node(T info)
    {
        this.info=info;
        this.next=null;
    }
}
public class QueueDemo2<T>
{
    public static <T> Node<T> insert(Node<T> rear, Node<T> front)
    {
        Scanner obj = new Scanner(System.in);
        System.out.print("Enter the value to insert in Queue : ");
        T element=(T) obj.nextInt();
        Node n = new Node<T>(element);
        if(rear==null || front==null)
        {
            front =n;
            rear= n;
        }
        else
        {
            rear.next=n;
            rear=n;
        }
        return front;
    }
}
```



## Question-2

```
public static <T> Node<T> delete(Node<T> rear, Node<T> front)
{
    if(front==null)
        System.out.println("Empty Queue,Nothing to Delete...");
    else
    {
        System.out.println(front.info+" is deleted...");
        front=front.next;
    }
    return front;
}
public static <T> void display(Node<T> rear, Node<T> front)
{
    Node<T> n = front;
    if(n==null)
        System.out.println("Queue is Empty...");
    else
    {
        System.out.println("Elements of Queue are : ");
        while(n!=null)
        {
            System.out.print(n.info+" <- ");
            n=n.next;
        }
        System.out.println();
    }
}
public static<T> Node<T> getRear(Node<T> front)
{
    if(front==null)
    {
        return null;
    }
    else
    {
        Node<T> n = front;
        while(n.next!=null)
        {
            n=n.next;
        }
        return n;
    }
}
```



## Question-2

```
public static void main(String args[])
{
    Scanner obj = new Scanner(System.in);
    Node<Object> front=null, rear=null;

    while(true)
    {
        System.out.println("****MENU****");
        System.out.println("0 : Exit");
        System.out.println("1 : Insert");
        System.out.println("2 : Delete");
        System.out.println("3 : Display");
        System.out.println("Enter your Choice: ");
        int choice=obj.nextInt();
        switch(choice)
        {
            case 0:
                System.exit(0);
                System.out.println("Thank you...");
                break;
            case 1:
                front=insert(rear, front);
                rear=getRear(front);
                break;
            case 2:
                front=delete(rear, front);
                break;
            case 3:
                display(rear, front);
                break;
            default:
                System.out.println("Wrong Choice");
        }
    }
}
```