Solution of Assignment 7

1. **Implementation of Queue using array (Static Data Structure)**

```
import java.util.*;

class static_Queue
{
	public static final int MAX = 4;
	public static int rear = -1;
	public static int front = -1;
	static int[] Q = new int[MAX];

	public static void insert()
	{
		Scanner sc = new Scanner(System.in);
		if(isFull())
			System.out.println("Queue Overflow, Insert not possible");
		else
		{
			if(front == -1)
			{
					front = 0;
			}
			System.out.println("Enter a data to be add: ");
			Q[++rear] = sc.nextInt();
		}
	 }

	public static void delete()
	{
		if(isEmpty())
			System.out.println("Queue underflow, delete not possible");
		else
		{
			System.out.println("Deleted item is: "+Q[front]);
			if(front == rear)
			{
				front = -1;
```

```
                rear = -1;
            }
            else
            {
                front++;
            }
        }
}

public static void display()
{
    if(front==-1)
        System.out.println("Empty Queue");
    else
    {
        System.out.println("Queue elements are: ");
        for(int i = front; i<=rear; i++)
        {
            System.out.print(Q[i]+" ");
        }
        System.out.println();
    }
}

public static boolean isFull()
{
    return rear==MAX-1;
}

public static boolean isEmpty()
{
    return front==-1;
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Program for creating Queue using array");
    while(true)
    {
```

```
                System.out.println("Menu for different operation");
                System.out.println("Press 0: Exit");
                System.out.println("Press 1: Insert");
                System.out.println("Press 2: Delete");
                System.out.println("Press 3: Display");
                System.out.println("Enter your choice: ");
                int choice = sc.nextInt();
                switch(choice)
                {
                        case 0: System.exit(0);
                        case 1: insert(); break;
                        case 2: delete(); break;
                        case 3: display();   break;
                        default: System.out.println("Wrong choice, try again");
                }
        }
    }
}
```

2. Implementation of Queue using linked list(Dynamic Data Structure)

```
import java.util.*;

class Node
{
    int data;
    Node next;
}

public class dynamic_Queue
{
    public static Node rear = null;
    public static Node front = null;

    public static void insert()
    {
        Scanner sc = new Scanner(System.in);
```

```
        Node p = new Node();
        System.out.println("Enter the data of new node: ");
        p.data = sc.nextInt();
        p.next = null;;
        if(rear == null)
        {
                rear = p;
                front = p;
        }
        else
        {
                rear.next = p;
                rear = p;
        }
}

public static void delete()
{
         if(front==null)
         {
                  System.out.println("Queue underflow, delete not possible");
         }
         else
         {
                  Node p = front;
                  front = front.next;
                  System.out.println("Deleted node info-- data value: "+p.data);
                  if(front == null)
                  {
                     rear = null;
                  }
         }
   }

public static void display()
 {
         if(front==null)
         {
                  System.out.println("Empty Queue");
         }
```

```
            else
            {
                Node p = front;
                System.out.println("Node details: \t value");
                 while(p!=null)
                 {
                        System.out.println(" \t\t "+p.data);
                        p = p.next;
                 }
            }
     }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Program for creating stack using linked list");
        while(true)
        {
            System.out.println("Menu for different operation");
            System.out.println("Press 0: Exit");
            System.out.println("Press 1: Insert");
            System.out.println("Press 2: Delete");
            System.out.println("Press 3: Display");
            System.out.println("Enter your choice: ");
            int choice = sc.nextInt();
            switch(choice)
            {
                case 0: System.exit(0);
                case 1: insert();  break;
                case 2: delete();   break;
                case 3: display();    break;
                default: System.out.println("Wrong choice, try again");
            }
        }
    }
}
```