Solution of Assignment 6

1. Implementation of stack using array (Static Data Structure)

```
import java.util.*;

class static_stack
{
	public static final int MAX = 5;

	public static int push(int S[], int top)
	{
		Scanner sc = new Scanner(System.in);
		if(isFull(top))
			System.out.println("Stack Overflow, Insert not possible");
		else
		{
			System.out.println("Enter a data to be add: ");
			top++;
			S[top] = sc.nextInt();
		}
		return top;
	}

	public static int pop(int S[], int top)
	{
		if(isEmpty(top))
			System.out.println("Stack underflow, delete not possible");
		else
		{
			System.out.println("Deleted item is: "+S[top]);
			top--;
		}
		return top;
	}

	public static void display(int S[], int top)
	{
		if(top==-1)
			System.out.println("Empty stack");
		else
		{
```

```
                int i = top;
                System.out.println("Array elements are: ");
                while(i>=0)
                {
                        System.out.print(S[i]+" ");
                        i--;
                }
                System.out.println();
        }
}

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

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

public static void main(String[] args)
{
        Scanner sc = new Scanner(System.in);
        System.out.println("Program for creating stack using array");
         int[] S = new int[MAX];
         int top = -1;
         while(true)
        {
                 System.out.println("Menu for different operation");
                System.out.println("Press 0: Exit");
                System.out.println("Press 1: Push");
                System.out.println("Press 2: Pop");
                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: top = push(S,top); break;
                        case 2: top = pop(S,top); break;
                        case 3: display(S,top);   break;
                        default: System.out.println("Wrong choice, try again");
                }
```

```
            }
        }
}
```

2. Implementation of stack using Linkedlist (Dynamic Data Structure)

```
import java.util.*;

class Node
{
   int info;
   Node next;
}

public class Stack
{
        public static Node top = null;

        public static void push()
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the number of new node: ");
                int data = sc.nextInt();
                Node node = new Node();
                node.info = data;
                node.next = top;
                top = node;
        }

        public static void pop()
        {
                 if(top==null)
                 {
                        System.out.println("Stack underflow, delete not possible");
                 }
                 else
                 {
                        Node q = top;
                        top = top.next;
                        System.out.println("Deleted node info-- data value: "+q.info);
                 }
           }
```

```
    public static void display()
    {
        if(top==null)
        {
            System.out.println("Stack underflow");
        }
        else
        {
            Node p = top;
            System.out.println("Node details: \t value");
            while(p!=null)
            {
                System.out.println(" \t\t "+p.info);
                p = p.next;
            }
        }
    }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Program for creating stack using linkedlist");
        while(true)
        {
            System.out.println("Menu for different operation");
            System.out.println("Press 0: Exit");
            System.out.println("Press 1: push");
            System.out.println("Press 2: pop");
            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: top = push();  break;
                case 2: top = pop();   break;
                case 3: display();    break;
                default: System.out.println("Wrong choice, try again");
            }
        }
    }
}
```