



CSE 2001: Data Structure & Algorithms
Programming Assignment-VI
(Stack)

```
//In this pdf 2 type of code in this..  
//using generics so that any type of element can insert inside the stack  
//Another one is simple stack only for Integer...
```



Question-1

```
import java.util.*;
public class StackDemo1
{
    public static int push(Object[] S, int top)
    {
        if (isFull(top))
            System.out.println("Stack is already full...Can't push anymore");
        else
        {
            System.out.print("Enter the element : ");
            Object element = (Object) obj.next();
            S[++top]=element;
        }
        return top;
    }
    public static int pop(Object[] S, int top)
    {
        if (isEmpty(top))
            System.out.println("Stack is empty...can't pop any element now");
        else
            System.out.println("Element "+ S[top--]+" is popped from the stack...");
        return top;
    }
    public static void display(Object[] S, int top)
    {
        if (isEmpty(top))
            System.out.println("Stack is empty. Nothing to display.");
        else
        {
            System.out.println("Stack elements are :");
            for (int i = top; i >= 0; i--)
                System.out.println(S[i]);
        }
    }
}
```



```
public static boolean isEmpty(int top)
{
    return (top == -1);
}
public static boolean isFull(int top)
{
    return top == MAX - 1;
}
public static final int MAX = 10;
static Scanner obj = new Scanner(System.in);
public static void main(String[] args)
{
    Object[] stack = new Object[MAX];
    int top = -1;
    while (true)
    {
        System.out.println("****MENU****");
        System.out.println("0: Exit");
        System.out.println("1: Push");
        System.out.println("2: Pop");
        System.out.println("3: Display");
        System.out.println("Enter your choice");
        int choice =obj.nextInt();
        switch (choice)
        {
            case 0:
                System.out.println("Thank you...Have a Great day...");
                System.exit(0);
                break;
            case 1:
                top=push(stack, top);
                break;
            case 2:
                top=pop(stack, top);
                break;
            case 3:
                display(stack, top);
                break;
            default:
                System.out.println("Invalid choice...");
                break;
        }
    }
}
```



Question-2

```
import java.util.Scanner;
public class StackDemo2
{
    static Scanner obj = new Scanner(System.in);
    static class Node<T>
    {
        T info;
        Node<T> next;
        Node(T element)
        {
            info = element;
            next = null;
        }
    }
    public static <T> Node<T> push(Node<T> top)
    {
        System.out.println("Enter the value:");
        T element = (T) obj.next();
        Node newNode=new Node<T>(element);
        newNode.next=top;
        top=newNode;
        return top;
    }
    public static <T> Node<T> pop(Node<T> top)
    {
        if (top == null)
            System.out.println("Stack is empty...can't pop any element now");
        else
        {
            System.out.println("Element " + top.info + " is popped from the
                                stack...");
            top = top.next;
        }
        return top;
    }
}
```



```
public static <T> void display(Node<T> top)
{
    if (top == null)
        System.out.println("Stack is empty , Nothing to display.");
    else
    {
        Node<T> current = top;
        System.out.print("Stack elements are :\n");
        int i=1;
        while (current != null)
        {
            System.out.println(i++ +" ->" +current.info);
            current = current.next;
        }
        System.out.println();
    }
}

public static void main(String[] args)
{
    Node<Object> top = null;
    while (true)
    {
        System.out.println("****MENU****");
        System.out.println("0: Exit");
        System.out.println("1: Push");
        System.out.println("2: Pop");
        System.out.println("3: Display");
        System.out.println("Enter your choice:");
        int choice = obj.nextInt();
        switch (choice) {
            case 0:
                System.exit(0);
            case 1:
                top=push(top);
                break;
            case 2:
                top = pop(top);
                break;
            case 3:
                display(top);
                break;
            default:
                System.out.println("Wrong choice");
        }
    }
}
```



//By Ansuman Swain

//Please ask your teacher which way to do

//If they agree to do with the first way then don't need to do the following code

//if they disagree with the first way then skip that and do the following code



```
//Don't Do it ....It is only for Practice and understanding...
import java.util.*;
public class StackDemo1withoutGenerics
{
    public static final int MAX = 10;
    public static int push(int S[], int top)
    {
        Scanner obj = new Scanner(System.in);
        if (isFull(top))
            System.out.println("Stack is already full... Cannot push any element
now...");  
        else
        {
            System.out.print("Enter the element to push:");
            int push = obj.nextInt();
            S[++top] = push;
            System.out.println("Element pushed successfully.");
        }
        return top;
    }
    public static int pop(int S[], int top)
    {
        if (isEmpty(top))
            System.out.println("Stack is empty. Cannot pop any element....");
        else
        {
            int x = S[top--];
            System.out.println("Popped element is " + x);
        }
        return top;
    }
    public static void display(int S[], int top)
    {
        if (isEmpty(top))
            System.out.println("Stack is empty...can't pop any element...");
        else
        {
            System.out.println("The elements in the Stack are :");
            for (int i = top; i >= 0; i--)
                System.out.println(S[i]);
        }
    }
    public static boolean isEmpty(int top)
    {
        if (top == -1)
            return true;
        return false;
    }
}
```



```
//Don't Do it ....It is only for Practice and understanding...
public static boolean isFull(int top)
{
    if(top == MAX -1)
        return true;
    return false;
}
public static void main(String[] args)
{
    Scanner obj = new Scanner(System.in);
    int stack[] = new int[MAX];
    int top = -1;
    while (true)
    {
        System.out.println("*** MENU ***");
        System.out.println("0: Exit");
        System.out.println("1: Push");
        System.out.println("2: Pop");
        System.out.println("3: Display");
        System.out.print("Enter your choice:");
        int choice = obj.nextInt();
        switch (choice)
        {
            case 0:
                System.out.println("Thank you...");
                System.exit(0);
            case 1:
                top = push(stack, top);
                break;
            case 2:
                top = pop(stack, top);
                break;
            case 3:
                display(stack, top);
                break;
            default:
                System.out.println("Invalid choice");
        }
    }
}
```



//Don't Do itIt is only for Practice and understanding...

Question-2:

```
import java.util.*;
class Node
{
    int info;
    Node next;

    Node(int data)
    {
        info = data;
        next = null;
    }
}
public class StackDemo2withoutGenerics
{
    public static void main(String args[])
    {
        Scanner obj = new Scanner(System.in);
        Node top=null;
        while(true)
        {
            System.out.println("****MENU****");
            System.out.println("0:Exit");
            System.out.println("1:Push");
            System.out.println("2:Pop");
            System.out.println("3:Display");
            System.out.println("Enter your choice: ");
            int choice=obj.nextInt();
            switch(choice)
            {
                case 0:
                    System.out.println("Thank you...");
                    System.exit(0);
                    break;
                case 1:
                    top=push(top);
                    break;
                case 2:
                    top=pop(top);
                    break;
                case 3:
                    display(top);
                    break;
                default:
                    System.out.println("Wrong choice");
            }
        }
    }
}
```



//Don't Do itIt is only for Practice and understanding...

Question-2:

```
public static Node push(Node top)
{
    Scanner obj = new Scanner(System.in);
    System.out.print("Enter the element to Push :");
    Node newNode = new Node(obj.nextInt());
    newNode.next=top;
    top=newNode;
    return top;
}
public static Node pop(Node top)
{
    if(top==null)
        System.out.println("Error : Stack UnderFlow");
    else
    {
        System.out.println("Element " + top.info + " popped from the stack.");
        top = top.next;
    }
    return top;
}
public static void display(Node top)
{
    if(top==null)
        System.out.println("Stack Empty");
    else
    {
        System.out.println("Elements of Stack");
        Node newNode=top;
        while(newNode!=null)
        {
            System.out.println(newNode.info);
            newNode=newNode.next;
        }
    }
}
```