```
package A6;

import java.util.Scanner;

/**
 *
 * @author Lakshmi Priya
 */

class Stack<T>{
    private T stack[];
    private int tos;

    public Stack(T stack[]){
        this.stack=stack;
        tos=-1;
    }

    public void push(T item){
        if(isFull()==true)
            System.out.println("\nStack is full!!\n");
        else{
            stack[++tos]=item;
        }
    }

    public T pop(){
        if(isEmpty()==true){
            System.out.println("\nStack is empty!!\n");
            return null;
        }
        else{
            return stack[tos--];
        }
    }

    public boolean isEmpty(){
        if(tos==-1)
            return true;
        return false;
    }

    public boolean isFull(){
        if(tos==stack.length-1)
            return true;
        return false;
    }

    public void clear(){
        tos=-1;
    }

    public void display(){
        System.out.println("Elements in stack: ");
        for(int i=tos;i>=0;i--){
            System.out.println(stack[i]);
```

```
        }
    }

}


public class GenStack {
    public static void main(String[] args) {
        int capacity=0,op;
        Scanner in=new Scanner(System.in);
        System.out.print("Enter capacity of stack: ");
        capacity=in.nextInt();

        System.out.println("****************************");
        System.out.println("STACK WITH INTEGER ELEMENTS");
        System.out.println("****************************");

        Stack<Integer> istack = new Stack<Integer>(new
Integer[capacity]);
        System.out.print("\nChoice of operation:\n\t1. Push\n\t2.
Pop\n\t3. Check if stack empty\n\t4. Check if stack full\n\t5.
Display stack\n\t0. Back\nEnter choice of operation: ");
        op=in.nextInt();
            while(op!=0){
                switch(op){
                    case 1: System.out.print("Enter item to be
pushed onto stack: ");
                            int i=in.nextInt();
                            istack.push(i);
                            break;
                    case 2: System.out.println("Popped item: " +
istack.pop());
                            break;
                    case 3: if(istack.isEmpty()==true)
                                System.out.println("Stack is
empty!!");
                            else
                                System.out.println("Stack is NOT
empty!!");
                            break;
                    case 4: if(istack.isFull()==true)
                                System.out.println("Stack is
full!!");
                            else
                                System.out.println("Stack is NOT
full!!");
                            break;
                    case 5: istack.display();
                            break;
                }
                System.out.print("\nChoice of operation:\n\t1.
Push\n\t2. Pop\n\t3. Check if stack empty\n\t4. Check if stack
full\n\t5. Display stack\n\t0. Back\nEnter choice of operation: ");
                op=in.nextInt();
            }
```

```
        System.out.println("*************************");
        System.out.println("STACK WITH FLOAT ELEMENTS");
        System.out.println("*************************");


        Stack<Float> fstack = new Stack<Float>(new Float[capacity]);
         System.out.print("\nChoice of operation:\n\t1. Push\n\t2.
Pop\n\t3. Check if stack empty\n\t4. Check if stack full\n\t5.
Display stack\n\t0. Back\nEnter choice of operation: ");
            op=in.nextInt();
            while(op!=0){
                switch(op){
                    case 1: System.out.print("Enter item to be
pushed onto stack: ");
                            float f=in.nextFloat();
                            fstack.push(f);
                            break;
                    case 2: System.out.println("Popped item: " +
fstack.pop());
                            break;
                    case 3: if(fstack.isEmpty()==true)
                                System.out.println("Stack is
empty!!");
                            else
                                System.out.println("Stack is NOT
empty!!");
                            break;
                    case 4: if(fstack.isFull()==true)
                                System.out.println("Stack is
full!!");
                            else
                                System.out.println("Stack is NOT
full!!");
                            break;
                    case 5: fstack.display();
                            break;
                }
                System.out.print("\nChoice of operation:\n\t1.
Push\n\t2. Pop\n\t3. Check if stack empty\n\t4. Check if stack
full\n\t5. Display stack\n\t0. Back\nEnter choice of operation: ");
                op=in.nextInt();
            }


        System.out.println("*************************");
        System.out.println("STACK WITH STRING ELEMENTS");
        System.out.println("*************************");


        Stack<String> sstack = new Stack<String>(new
String[capacity]);
         System.out.print("\nChoice of operation:\n\t1. Push\n\t2.
Pop\n\t3. Check if stack empty\n\t4. Check if stack full\n\t5.
Display stack\n\t0. Back\nEnter choice of operation: ");
            op=in.nextInt();
```

```
            while(op!=0){
                switch(op){
                    case 1: System.out.print("Enter item to be
pushed onto stack: ");
                            in.nextLine();
                            String s=in.nextLine();
                            sstack.push(s);
                            break;
                    case 2: System.out.println("Popped item: " +
sstack.pop());
                            break;
                    case 3: if(sstack.isEmpty()==true)
                                System.out.println("Stack is
empty!!");
                            else
                                System.out.println("Stack is NOT
empty!!");
                            break;
                    case 4: if(sstack.isFull()==true)
                                System.out.println("Stack is
full!!");
                            else
                                System.out.println("Stack is NOT
full!!");
                            break;
                    case 5: sstack.display();
                            break;
                }
                System.out.print("\nChoice of operation:\n\t1.
Push\n\t2. Pop\n\t3. Check if stack empty\n\t4. Check if stack
full\n\t5. Display stack\n\t0. Back\nEnter choice of operation: ");
                op=in.nextInt();
             }

    }
}
```