//By Ansuman Swain

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 on e is simple stack only for Integer...
```

//By Ansuman Swain

```
package Assignment_5;
import java.util.Scanner;
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.println("Enter the type of element to push:");
            System.out.println("1. Integer");
            System.out.println("2. Character");
            System.out.println("3. String");
            int type = obj.nextInt();
            switch (type)
            {
                case 1:
                    System.out.print("Enter an integer: ");
                    int intelement = obj.nextInt();
                    S[++top] =intelement;
                    break;
                case 2:
                    System.out.print("Enter a character: ");
                    char charelement = obj.next().charAt(0);
                    S[++top] =charelement;
                    break;
                case 3:
                    System.out.print("Enter a string: ");
                    String stringElement = obj.next();
                    S[++top] = stringElement;
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
                    break;
            }
        }
        return top;
    }
```

//By Ansuman Swain

```
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;
    }
```

```
//By Ansuman Swain
```

```java
    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;
            }
        }
    }
}
```

//By Ansuman Swain

```
package Assignment_5;
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 type of element to push:");
        System.out.println("1. Integer");
        System.out.println("2. Character");
        System.out.println("3. String");
        int type = obj.nextInt();
        System.out.println("Enter the value:");
        switch(type)
        {
        case 1:
            int intValue = obj.nextInt();
            Node intNode = new Node<>(intValue);
            intNode.next = top;
            top = intNode;
            break;
         case 2:
            char charValue = obj.next().charAt(0);
            Node charNode  = new Node<>(charValue);
            charNode.next = top;
            top = charNode;
            break;
```

//By Ansuman Swain

```
case 3:
            String stringValue = obj.next();
            Node stringNode = new Node<>(stringValue);
            stringNode.next = top;
            top = stringNode;
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
            break;
        }
        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();
        }
    }
```

```
//By Ansuman Swain
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
```

//By Ansuman Swain

```
package Assignment_5;
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]);
        }
    }
```

//By Ansuman Swain

```
public static boolean isEmpty(int top)
    {
        if(top == -1)
            return true;
        return false;
    }
    public static boolean isFull(int top)
    {
        if(top == MAX -1)
            return true;
        return false;
    }
    public static void main(String[] args)
    {
        Scanner sc = 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 = sc.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");
          }
      }
   }
}
```

```
//By Ansuman Swain
```

```java
package Assignment_5;
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");
            }
        }
    }
```

//By Ansuman Swain

```
public static boolean isEmpty(int top)
    {
        if(top == -1)
            return true;
        return false;
    }
    public static boolean isFull(int top)
    {
        if(top == MAX -1)
            return true;
        return false;
    }
    public static void main(String[] args)
    {
        Scanner sc = 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 = sc.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");
          }
      }
   }
}
```