Programming Paradigms Laboratory

B.Tech.

Name : Subhendu Maji

Roll Number : 18ETCS002121

Department : Computer Science and Engineering

Faculty of Engineering & Technology

Ramaiah University of Applied Sciences

Faculty	Engineering & Technology
Programme	B. Tech. in Computer Science and Engineering
Year/Semester	2nd Year / 4th Semester
Name of the Laboratory	Programming Paradigms Laboratory
Laboratory Code	19CSL217A

Name: Subhendu Maji Reg. No.: 18ETCS002121

Laboratory 8

Title of the Laboratory Exercise: Interface and Exception handling

1. Questions

    a. Develop a java interface for Stack ADT and implement it using class Array.

    b. Develop a Java program using exception handling to input five array elements through command line and find the sum and average by throwing ArrayIndexOutOfBoundException.

    c. Develop a Java program using exception handling to throw NumberFormatException using Integer.parseInt to find sum of two input strings by typecasting them into integers.

2. Calculations/Computations/Algorithms

```java
package interfaces;
interface stack{
    public abstract void push(int b);
    public abstract void pop();
    public abstract void display();
}
class Array implements stack
{
    static int a[]=new int[5];
    static int top=-1;
    public void push(int b)
    {
        top++;
        if(top>5)
        {
            System.out.println("overflow");
        }
        else
        {
            System.out.println("Element pushed : "+b);
            a[top]=b;
        }
    }
    public void pop()
    {
        if(top==-1)
        {
            System.out.println("stack is overflowing !");
        }
        else
        {
            int x=a[top];
            top--;
            System.out.println("Element has been popped:" +x);
        }
    }
    public void display()
    {
        System.out.println("Stack elements are:");
        for(int i=top;i>0;i--)
        {
            System.out.println(a[i]);
        }
    }
}
```

```
public class Interfaces {

    public static void main(String[] args) {
        Array s=new Array();
        s.push(1);
        s.push(2);
        s.push(3);
        s.pop();
        s.display();
    }
}
```

Figure 8.1 Represents stack ADT using array

```
package strings;
public class Strings {
    public static void main(String[] args) {
        try{
    String str1= "ten";
    String str2= "eleven";
    int x = Integer.parseInt(str1);

        String sum = str1+str2;


      int y = Integer.parseInt(str2);
      System.out.println(x+y);
    }

    catch(Exception e)
    {
      System.err.println("Unable to format. " + e);
    }
  }
}
```

Figure 8.2 Represents exception handling to input five array elements through command line.

```
package sum;
public class Sum {
    public static void main(String[] args) {
         try
        {
            int sum=0;
            int avg;

        for (int i=0;i<5;i++)
        {
            System.out.println(args[i]);
            sum=sum+Integer.parseInt(args[i]);
        }
        System.out.println("sum="+sum);
        avg=sum/5;
        System.out.println("avg="+avg);

    }
    catch(Exception e)
    {
        System.out.println(e);
    }
        finally {
        System.out.println("success");
    }

}
}
```

Figure 8.3 Represents exception handling to throw NumberFormatException using Integer.parseInt

3. Presentation of Results

```
Element pushed : 1
Element pushed : 2
Element pushed : 3
Element has been popped:3
Stack elements are:
2
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 8.4 output for the stack ADT

```
1
2
3
4
5
sum=15
avg=3
success
BUILD SUCCESSFUL (total time: 1 second)
```

Figure 8.5 Represents the output without arrayoutofboundsexception

```
1
2
3
java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
success
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 8.6 Represents the output with arrayoutofboundsexception

```
compile:
run:
Unable to format. java.lang.NumberFormatException: For input string: "ten"
BUILD SUCCESSFUL (total time: 2 seconds)
```

Figure 8.7 Represents the output for numberformatexception

4. Conclusions

   If a request for a negative or an index equal. To size of an array is made then the java throws a array index out of bounds.

   Number format exception usually occurs when you try to do something like convert a string to a numeric value.

5. Limitations of Experiments and Results

The array index out of bounds exception is a runtime exception thrown only at a runtime.

The complier doesn't check for this error during the compilation of a program.