Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab no. :	11
Student name:	Pooja Jagdish Verma
Roll no. :	21

Title:-

1. Write a program in java if a number is less than 0 and greater than 10 it generates the user-defined exception "out of range". Else it displays the square of the number.
2. Write a program in java to enter the number. If the first and second number is not entered it will generate the exception. Also, divide the first number with the second number and generate the arithmetic exception.

Learning Objective:

Students will be able to implement user-defined exceptions

Learning Outcome:

Understanding the exception handling concept and making the programming interface error-free.

Course Outcome:

ECL304.3	Articulate exception handling methods.

Theory:

Q1. What is exception handling and how is it achieved in JAVA?

Exception Handling in Java is a distinctive approach to improvise a Java application's convenience and performance capabilities. Exceptions, if not handled properly, may pose a severe threat to the application program in real-time.
Types of Exceptions: There are technically two types of exceptions. The third variety is am error. They are:
1. Checked Exception: The class that inherit all the exceptions from the throwable parent class directly, but except for the run-time exception, are called the checked exception.
2. Unchecked Exceptions: The classes that inherit only the run time exceptions are called as the unchecked exceptions.
3. Errors: An unrecoverable event that collapses the entire application program

Q2. Explain user defined exceptions in java?

User Defined Exception or custom exception is creating your own exception class and throws that exception using ‘throw’ keyword. This can be done by extending the class Exception.

There is no need to override any of the above methods available in the Exception class, in your derived class. But practically, you will require some amount of customizing as per your programming needs.
The keyword **throw** is used to create a new Exception and throw it to the catch block.

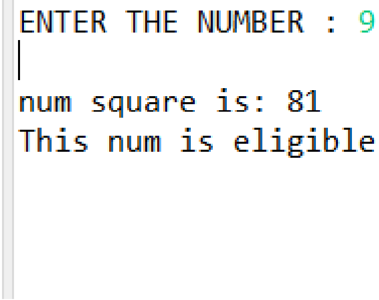
Algorithm :	**STEP 1: Start** **STEP 2: Exception extend class OutofRange** **STEP 3:Write attributes of class** **STEP 4: Use try and catch for the numbers** **STEP 5:If the number is >10, say out of range** **STEP 6:If the number is <11, show its square at OP** **STEP 7:Print OP** **STEP 8:Stop**

Program:

```
import java.util.*;


class OutOfRange extends Exception{
   int num;
   OutOfRange(int a){
      num = a;
   }


   public String toString()
   {
      return ("num is out of range: "+ num);
   }


}


class Main{
```

```
void test(int num)
{
    try{
        if(num<0||num>10)
            throw new OutOfRange(num);


        System.out.println();
        System.out.print("num square is: ");
        System.out.println( num*num);
    }
    catch(OutOfRange u)
    {
        System.out.println("Out of range ");
        u.printStackTrace();
        System.out.println("This num is not eligible");
        System.exit(0);
    }
    System.out.println("This num is eligible ");
}

public static void main(String args[])
{
    int num;

    Scanner sc = new Scanner(System.in);

    System.out.print("ENTER THE NUMBER : ");
    num = sc.nextInt();
```

	Main e = new Main();
	e.test(num);
	}
	}

Input given:	9
Output Screenshot:	 ENTER THE NUMBER : 9 num square is: 81 This num is eligible

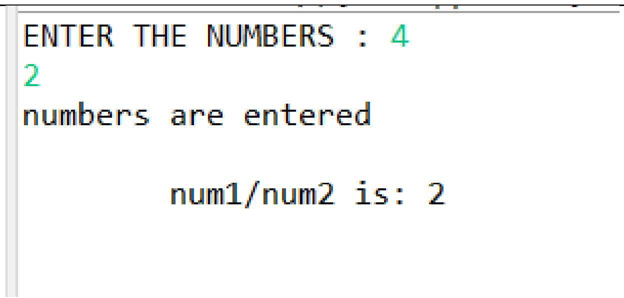
Algorithm :	STEP 1: Start STEP 2:Create class IsNum and extend it to exception STEP 3: If string is given at IP, say the number should be an ninteger STEP 4: In main class, use try and catch for dividing the number with the other number. STEP 5:If exception is created, display cannot be divided by 0 STEP 6:Create main class, write the code for throw and catch as valid/ invalid nos, make its objects STEP 7:Display the OP STEP 8:Stop

Program:	import java.io.*;
	import java.util.Scanner;
	class IsNum extends Exception{

```
    public String toString()
    {
        return ("number is not valid it should be an integer : ");
    }


}


class Main{


    void test(int num1,int num2)
    {
        try{


            int res=num1/num2;
            System.out.println();
            System.out.print("        num1/num2 is: ");
            System.out.println(res);
        }
        catch(ArithmeticException e)
        {
            System.out.println(" can't divide by zero "+e);
        }


    }


    public static void main(String args[])
    {
```

```
int num1=0,num2=0;


Scanner sc = new Scanner(System.in);


System.out.print("ENTER THE NUMBERS : ");
try
{
    if(sc.hasNextInt())
    {
        num1=sc.nextInt();
    }
    else
    {
        throw new IsNum();
    }
    if(sc.hasNextInt())
    {
        num2=sc.nextInt();
    }
    else
    {
        throw new IsNum();
    }
}


catch(IsNum u)
{
    System.out.println(" INVALID ");
```

	u.printStackTrace();
	System.out.println("his number is not entered");
	System.exit(0);
	}
	
	System.out.println("numbers are entered ");
	Main e = new Main();
	e.test(num1,num2);
	}
	}

Input given:	**First number: 4** **Second number: 2**
Output Screenshot:	 ENTER THE NUMBERS : 4 2 numbers are entered num1/num2 is: 2