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 Number:	**11**
Student Name:	**Raveena Rupesh Pitale**
Roll No :	**27**

Title:

1. Write a program injava 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:

What is exception handling and how is it achieved in JAVA?
Java exception handling is managed via five keywords: try, catch, throw, throws, and finally.

Any exception that is thrown out of a method must be specified as such by a throws clause.

Any code that absolutely must be executed after a try block completes is put in a finally block.

It is one of the powerful mechanism to handle the runtime errors so that the normal flow of the application can be maintained. There are mainly two types of exceptions in Java as follows: Checked exception, Unchecked exception.

The core advantage of exception handling is to maintain the normal flow of the application.

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.
There is no need to override any of the above methods available in the Exception class, in your derived class.
You can create your own exceptions in Java:-

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

1.All exceptions must be a child of Throwable.
2.If you want to write a checked exception that is automatically enforced by the Handle or Declare Rule, you need to extend the Exception class.
3.If you want to write a runtime exception, you need to extend the RuntimeException class.

An exception is a problem that arises during the execution of the program. In Object-Oriented Programming language, Java provides a powerful mechanism to handle such exceptions. Java allows to create own exception class, which provides own exception class implementation.

Algorithm :01	**1. Start** **2. Create Outofrange class.** **3. Create the main class to take input of data and perform the operation.** **4. Write the exception cases i.e. the try catch function** **5. End**
Program:01	

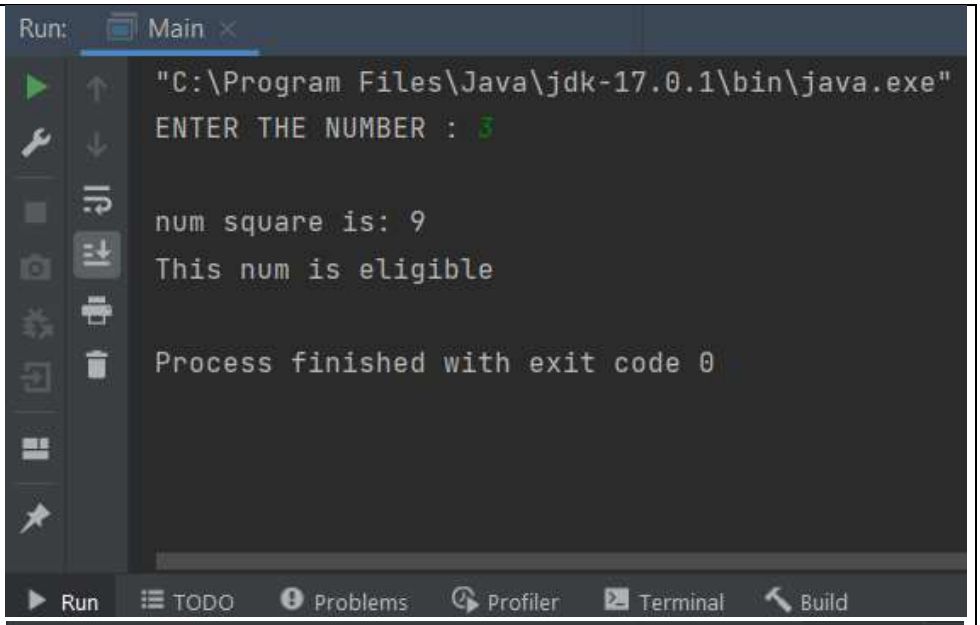
```
package com.company;
import java.util.*;

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);
```

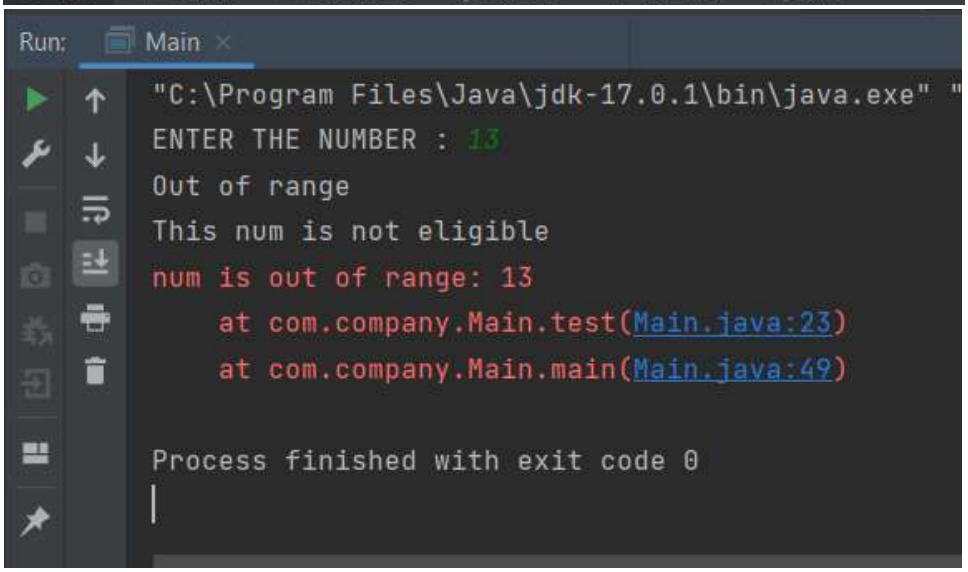
Faculty: Ms. Deepali Kayande

```
        }
        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:01	**Input 1:- 3** **Input 1:- 13**

Output Screenshot:01





Algorithm :02

1. **Start**
2. **Create Isnum class.**
3. **Create the main class to take input of data and perform the operation.**
4. **Write the exception cases i.e. the try catch function**
5. **End**

Program:02

```
package com.company;
import java.io.*;
import java.util.Scanner;
class IsNum extends Exception{

    public String toString()
    {
```

Faculty: Ms. Deepali Kayande

```
        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:02

Num 1 :-5

Num 2 :-10

Output Screenshot:02