Lab Number:	11
Student Name:	Nitish Chavan
Roll No :	18

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:

●   What is exception handling and how is it achieved in JAVA?

The **Exception Handling in Java** is one of the powerful *mechanism to handle the runtime errors* so that the normal flow of the application can be maintained. Java exception handling is managed via five keywords: try, catch, throw, throws, and finally. Program statements that you think can raise exceptions are contained within a try block. If an exception occurs within the try block, it is thrown. Your code can catch this exception (using catch block) and handle it in some rational manner. System-generated exceptions are automatically thrown by the Java run-time system. To manually throw an exception, use the keyword throw. 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.

●   Explain user defined exceptions in java?

Java user-defined exception is a custom exception created and throws that exception using a keyword 'throw'. It is done by extending a class 'Exception'. 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

Faculty: Ms. Deepali Kayande

provides own exception class implementation. Such exceptions are called user-defined exceptions or custom exceptions.

Algorithm 1:	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. thetry catch function
	5. End
Program 1:	```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)

  {
``` |

```java
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 ");
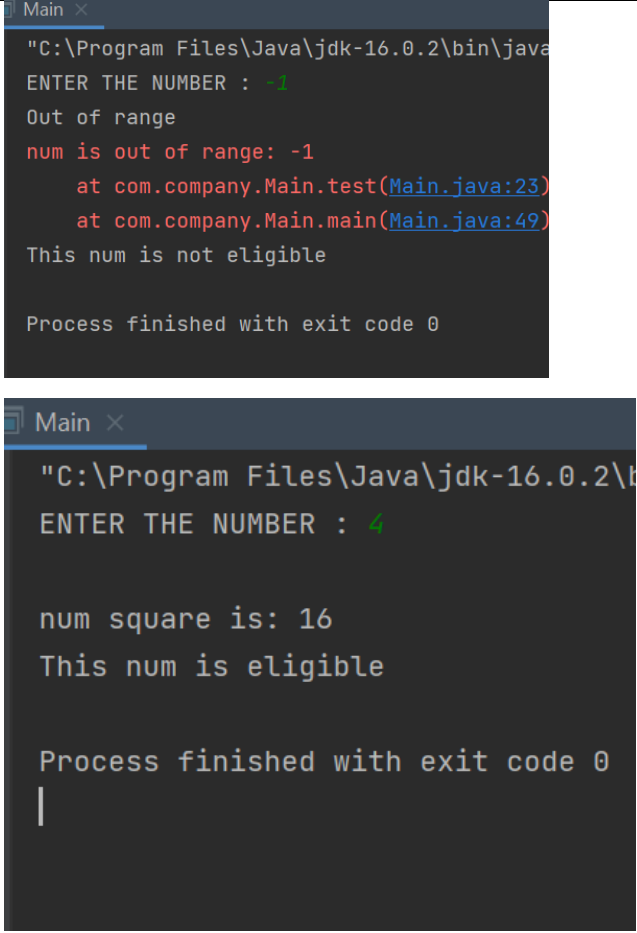}


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 1:	**1 input: -1** **2input: 4**
Output Screenshot 1:	
Algorithm 2:	6. **Start** 7. **Create Isnum class.** 8. **Create the main class to take input of data and perform the operation.** 9. **Write the exception cases i.e. thetry catch function** 10. **End**
Program 2:	**package com.company;**

```java
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);

    }


  }
```

```java
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 2:	**Num 1 – 4** **Num 2 - 5**
Output Screenshot 2:	Main × "C:\Program Files\Java\jdk-16.0.2\bin\j ENTER THE NUMBERS : 4 5 numbers are entered num1/num2 is: 0 Process finished with exit code 0

Faculty: Ms. Deepali Kayande