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:	**Sakshi Vadiraj Kaveri**
Roll No :	**33**

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?

An exception is a problem that arises during the execution of a program. It can occur for various reasons say-

1) A user has entered an invalid data.
2) File not found.
3) A network connection has been lost in the middle of communications.
4) The JVM has run out of a memory.

Handling an exception is very important, else it leads to system failure. But how do you handle these exceptions? . Java provides various methods to handle the Exceptions like:

1) Try- The try block contains a set of statements where an exception can occur. It is always followed by a catch block, which handles the exception that occurs in the associated try block.

2) Catch- A catch block is where you handle the exceptions. This block must follow the try block and a single try block can have several catch blocks associated with it. You can catch different

exceptions in different catch blocks. When an exception occurs in a try block, the corresponding catch block that handles that particular exception executes.

3) Finally- A finally block contains all the crucial statements that must be executed whether an exception occurs or not. The statements present in this block will always execute, regardless an exception occurs in the try block or not.

4) Throw- Used to explicitly throw an exception, it is followed by an instance. It is used within a method and it Cannot throw multiple exceptions.

5) Throws- Used to declare an exception it is followed by a class. It is used with a method signature and it Can declare multiple exceptions.

- Explain user defined exceptions in java?

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

Exception is nothing but an object of an exception class. We can create our own exception in the following ways:

1) By creating a class which extends the superclass of all the exception classes -Exception class.
   OR
2) By creating a class which extends any of the subclasses of the Exception class, such as IOException etc.

After creating our own exception class, we will be able to throw and catch our exception just as we throw and catch predefined checked/unchecked exceptions.

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.

ALGORITHM:

1) In main method throws keyword is used to declare exception InvalidNumber.
2) Declare variable n and take input from user.

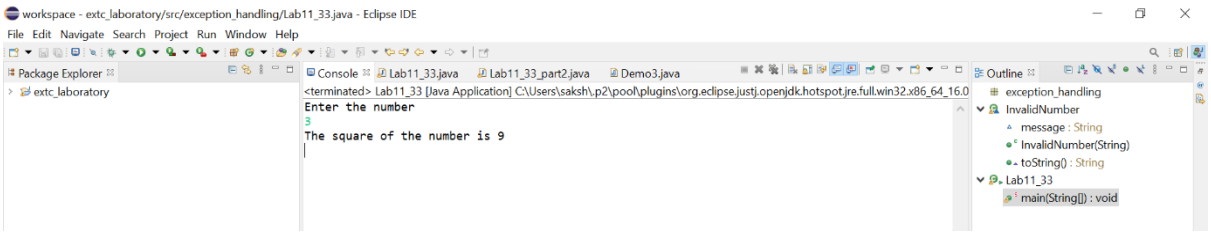
3) In the try block throw exception when the number entered is less than zero or greater than 10.
4) In catch block execute the code to be displayed when exception occurred.

PROGRAM:

```
/*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.
 */
package exception_handling;

import java.util.Scanner;
class InvalidNumber extends Exception{
	String message;
	public InvalidNumber(String message) {
	  this.message = message;
    }

    @Override
    public String toString() {
        return message;
    }
}

public class Lab11_33{
	public static void main(String args[]) throws InvalidNumber {
		Scanner s=new Scanner(System.in);
		int n;
		System.out.println("Enter the number ");
		n=s.nextInt();
		try {
			if(n<0) {
				throw new InvalidNumber("The number is less than zero
so it's not valid");
			}
			else if(n>10) {
				throw new InvalidNumber("The number is greater than 10
so it's not valid");
			}
			else {
				int calc=n*n;
				System.out.println("The square of the number is
"+calc);
			}
		}
		catch (InvalidNumber ob) {
			ob.printStackTrace();
		}
	}
}
```

INPUT GIVEN: 3,-1,20

OUTPUT SCREENSHOT:



3. 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.

ALGORITHM:

1) In main method throws keyword is used to declare exception Error.

2) In try block declare variable i and j and initialize variable k and take input from user.

3) in catch block the arithmetic exception is handled and the message (error/exception message) is displayed on the screen.

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

PROGRAM:

```
/*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.
 */
package exception_handling;

import java.util.Scanner;
class Error extends Exception{
	String message;
	public Error(String message) {
	  this.message = message;
    }

    @Override
    public String toString() {
        return message;
    }
}


public class Lab11_33_part2 {

	public static void main(String[] args) throws Error {
		try {

		Scanner sc=new Scanner(System.in);
		int i,j;
		float k;
		System.out.println("Enter a first and second number");
		i=sc.nextInt();
		j=sc.nextInt();
		k=(float) (i/j);
		System.out.println("The value of output is " +k);
		//throw new Error("The numbers are not entered");


	}
		catch(ArithmeticException e) {
			System.out.println("Can not divide by 0 "+e);
		}

}
}
```

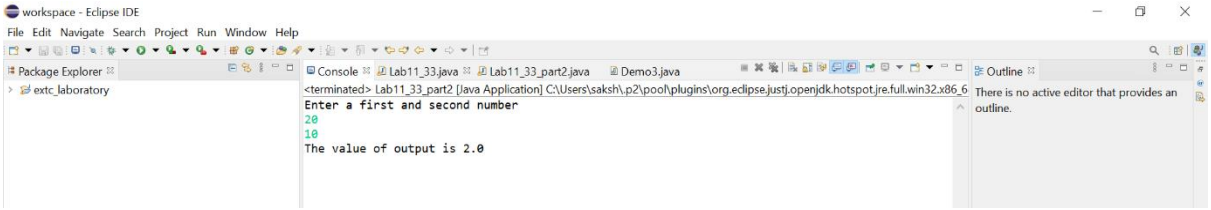
Input given:

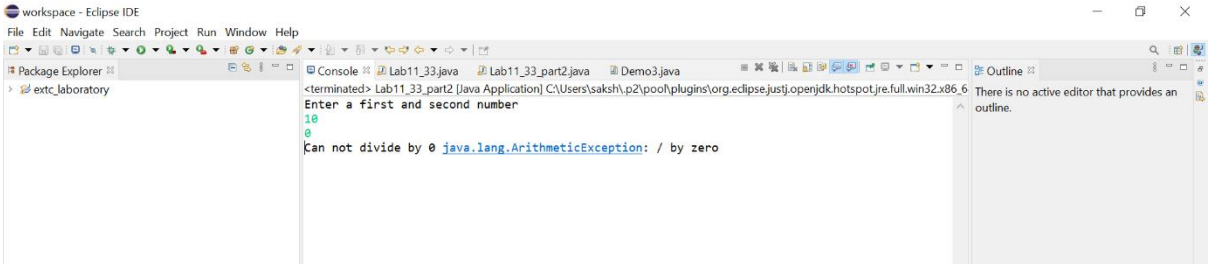
i=20, j=10;

i=10, j=0

Output screenshot:

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





Faculty: Ms. Deepali Kayande