**PRACTICAL – 5(2)**

**Aim: A piece of Java code is given below. You have to complete the code by**

**writing down the handlers for exceptions thrown by the code. The**

**exceptions the code may throw along with the handler message are listed**

**below:**

**Division by zero: Print "Invalid division".**

**String parsed to a numeric variable: Print "Format**

**mismatch". Accessing an invalid index in string: Print**

**"Index is invalid".**

**Accessing an invalid index in array: Print "Array index is invalid".**

**MyException: This is a user defined Exception which you need to create. It**

**takes a parameter param. When an exception of this class is encountered, the**

**handler should print "MyException[param]", here param is the parameter passed to the exception class.**

**Exceptions other than mentioned above: Any other exception except**

**the above ones fall in this category. Print "Exception encountered".**

**Finally, after the exception is handled, print "Exception**

**Handling Completed".**

**Example: For an exception of MyException class if the parameter value is**

**5, the message will look like**

**MyException[5].**

**SOURCE CODE:**

import java.util.\*;

class myexception extends Exception {

    public myexception(String *s*) {

        super(*s*);

    }

}

public class Practical\_5\_2 extends Exception {

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

*int* a, b, ans;

        Scanner sc = new Scanner(System.in);

        try {

            a = sc.nextInt();

            b = sc.nextInt();

            ans = a / b;

            String s = new String();

            s = "Name";

            System.out.println("Answer: " + ans);

            System.out.println(s.charAt(2));

            sc.close();

            throw new myexception("hello");

        } catch (myexception E) {

            System.out.println(E.getMessage());

            System.out.println("My Exception[" + E.getMessage() + "]");

        } catch (ArithmeticException e) {

            System.out.println("invalid division");

        } catch (InputMismatchException e) {

            System.out.println("format mismatch");

        } catch (StringIndexOutOfBoundsException e) {

            System.out.println("Index is invalid");

        } catch (ArrayIndexOutOfBoundsException e) {

            System.out.println("Array index is invalid");

        } finally {

            System.out.println("Exception Handling completed");

        }

        System.out.println("\n20DCE019 - Yatharth Chauhan");

    }

}

**OUTPUT:**

**Text

Description automatically generated**

**CONCLUSION:** In this program earnt how to handle exceptions using nested try and catch block along with finally block.