**Assignment: Exception Handling**

**Program 1: Write a small piece of code which shows simple usage of try-catch block with throw and throws keyword.**

public class \_01\_throw\_and\_throws {

    public static void main(String[] args) {

        try{

            printNum(10);

            printNum(200);

        }

        catch(Exception e){

            e.printStackTrace();

        }

    }

    public static void printNum(int n) throws Exception {

       if(n>100){

        throw new Exception("Exception thrown");

       }

       else{

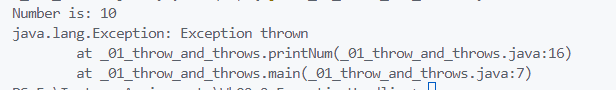
        System.out.println("Number is: "+n);

       }

    }

}

**Output:**

****

**Program 2: Write code to throw a custom exception when entered number is greater than 100 or less than 0.**

import java.util.Scanner;

public class \_02\_CustomException {

    private static class NumberOutOfoundException extends RuntimeException{

        NumberOutOfoundException(String s){

            super(s);

        }

    }

    public static void main(String[] args) {

        System.out.println("Enter a number: ");

        Scanner sc=new Scanner(System.in);

      int num=sc.nextInt();

     try{

        if(isNumOutOfBounds(num)){

            throw new NumberOutOfoundException("Number out of bounds");

        }

        System.out.println("Number is: "+num);

     }

     catch(NumberOutOfoundException e){

        e.printStackTrace();

     }

      finally{

          sc.close();

      }

    }

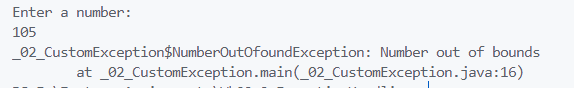
    private static boolean isNumOutOfBounds(int num) {

        return num>100 || num<0;

    }

}

**Output:**

****

**Program 3: Program to demonstrate chained exceptions**

import java.util.Scanner;

public class \_03\_ChainedExceptions {

    private static class ApiException extends RuntimeException {

        ApiException(String m) {

            super(m);

        }

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter two numbers");

        int n = sc.nextInt();

        int m = sc.nextInt();

        try {

            int sum = calcSum(n, m);

            System.out.println("Quotient: " + sum);

        } catch (ArithmeticException e) {

            ApiException apiException = new ApiException("Error occurred while performing division operation");

            apiException.initCause(e);

            throw apiException;

        }

        finally{

            sc.close();

        }

    }

    private static int calcSum(int a, int b) {

      return a/b;

    }

}

****