**Exceptions**

**1. Write a program to generate Arithmetic Exception without exception handling**

public class ArithmeticException

{

void divide(int a, int b)

{

// performing divison and storing th result

int res = a / b;

System.out.println("Division process has been done successfully.");

System.out.println("Result came after division is: " + res);

}

// main method

public static void main(String argvs[])

{

// creating an object of the class ArithmeticException

ArithmeticException obj = new ArithmeticException();

obj.divide(1, 0);

}

}

**2. Handle the Arithmetic exception using try-catch block**

class Main {

public static void main(String[] args) {

try {

// code that generate exception

int divideByZero = 5 / 0;

System.out.println("Rest of code in try block");

}

catch (ArithmeticException e) {

System.out.println("ArithmeticException => " + e.getMessage());

}

}

}

**3. Write a method which throws exception, Call that method in main class without try block**

import java.io.IOException;

class Testthrows1{

void m()throws IOException{

throw new IOException("device error");//checked exception

}

void n()throws IOException{

m();

}

void p(){

try{

n();

}catch(Exception e){System.out.println("exception handled");}

}

public static void main(String args[]){

Testthrows1 obj=new Testthrows1();

obj.p();

System.out.println("normal flow...");

}

}

**4. Write a program with multiple catch blocks**

public class MultipleCatchBlock1 {

public static void main(String[] args) {

try{

int a[]=new int[5];

a[5]=30/0;

}

catch(ArithmeticException e)

{

System.out.println("Arithmetic Exception occurs");

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("ArrayIndexOutOfBounds Exception occurs");

}

catch(Exception e)

{

System.out.println("Parent Exception occurs");

}

System.out.println("rest of the code");

}

}

**7. Write a program with finally block**

class TestFinallyBlock {

public static void main(String args[]){

try{

//below code do not throw any exception

int data=25/5;

System.out.println(data);

}

//catch won't be executed

catch(NullPointerException e){

System.out.println(e);

}

//executed regardless of exception occurred or not

finally {

System.out.println("finally block is always executed");

}

System.out.println("rest of phe code...");

}

}

**8. Write a program to generate Arithmetic Exception**

public class ArithmeticExceptionTest {

public static void main(String[] args) {

int a = 0, b = 10;

int c = b/a;

System.out.println("Value of c is : "+ c);

}

}

**10. Write a program to generate ClassNotFoundException**

public class GFG {

public static void main(String args[])

{

try {

Class.forName("GeeksForGeeks");

}

catch (ClassNotFoundException ex) {

ex.printStackTrace();

}

}

}