**1.** public class DivideZero {

static int anyFunction (int x, int y ){

try {

int a = x/y;

return a;

}

catch (ArithmeticException e) {

System.out.println ( "Division by zero" );

}

return 0;

}

public static void main (String args[]) {

int a,b, result;

a=0;

b=0;

System.out.print("Enter any two integers : ");

try{

a = System.in.read();

b = System.in.read();

}catch(Exception e){}

result = anyFunction (a, b);

System.out.println ( "Result : " + result);

}

}

**Find out the types of exceptions.**

**Unchecked Exception (Arithmetic Exception- DivideByZero Exception)**

**2.**

class CommandLineInput {

public static void main (String args[ ] {

int number, InvalidCount = 0; validCount = 0;

for (int i = 0; i < args.length; i++)

{

try {

number = Integer.parseInt(args[i]);

} catch (NumberFormatException e)

{

inavlidCount++;

System.out.println ( �Invalid number at � + i + args.[i]);

}

validCount++;

System.out.println ( �Valid number at � + i + args.[i]);

}

System.out.println ( �Invalid entries: � + inValidCount);

System.out.println ( �Valid entries: � + validCount);

}

}

}

**Find out the types of exceptions.**

**Unchecked Exceptions (NumberFormat Exception)**

**3.**

public class MultiCatch {

public static void main (String args[ ]) {

try {

int i = args.length; // No of arguments in the command line

String myString[] = new String[i];

// If i = 0 then myString null pointer error

// #1 // if(myString[0].equals(�Java�));

System.out.println("First word is Java !");

System.out.println( " Number of arguments = " + i );

// # 2 // int x = 18/ i;

int y[ ] = {555, 999};

// y is an array of size 2 and index are 0,1

// #3 // y[ i ] = x;

// Index is out-of-range may occur if i > 1

}

catch (ArithmeticException e ) { // To catch the error at #2

System.out.println ( " Div by 0 : "+ e );

}

catch (NullPointerException e ) {// To catch the error at #1

System.out.println ( "A null pointer exception :" + e );

}

catch (ArrayIndexOutOfBoundsException e ) {

// To catch the error at #3

System.out.println ("Array Index OoB : " + e);

}

}

}

**Find out the types of exceptions.**

**Unchecked Exceptions –(ArithmeticException,NullPointerException,ArrayIndexOutOfBoundsException)**

**4.**

import java.lang.\*;

public class exceptions{

public static void main(String Args[]) throws Exception{

int[] array = new int[3];

try{

for (int i=0;i<4;++i)

{

array[i] = i;

}

System.out.println(array);

}

catch(ArrayIndexOutOfBoundsException e){

System.out.println("filIn: " + e.fillInStackTrace());

System.out.println("cause: " + e.getCause());

System.out.println("local: " + e.getLocalizedMessage());

System.out.println("messa: " + e.getMessage());

System.out.println("trace: " + e.getStackTrace());

System.out.println();

System.out.println();

System.out.print("trace: "); e.printStackTrace();

System.out.println();

System.out.print("string: ");e.toString();

System.out.println();

System.out.println();

//printed just to inform that we have entered the catch block

System.out.println("Oops, we went too far, better go back to 0!");

throw (Exception) new Exception().initCause(e);

}

finally{

System.out.println(array);

//method call to continue program

}

}

}

**Find out the types of exceptions.**

**Unchecked Exception – ArrayIndexOutOfBoundsException.**

**5.**

class ExceptionTest {

public static int j;

public static void main (String args[ ] ) {

for (int i = 0; i < 4; i++ ) {

try {

switch (i) {

case 0 :

int zero = 0;

j = 999/ zero; // divide by zero

break;

case 1:

int b[ ] = null;

j = b[ 0] ; // Null pointer error

break;

case 2 :

int c[] = new int [2] ;

j = c[10]; // Array index is out-of-bound

break;

case 3 :

char ch = "Java".charAt(9) ; // String index is out-of-bound

break;

} // switch

} // try

catch (Exception e) { // To catch an exception

System.out.println "In Test case # " + i + "\n" );

System.out.println (e) ;

} // catch

} // main

} // class

}

**Find out the types of exceptions.**

**Unchecked Exception – (ArithmeticException,NullPointerException,ArrayIndexOutOfBoundsException)**