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

class Example1

{

public static void main(String args[])

{

try{

int num1=30, num2=0;

int output=num1/num2;

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

}

catch(ArithmeticException e){

System.out.println ("You Shouldn't divide a number by zero");

}

}

}

**Output of above program:**

You Shouldn't divide a number by zero

**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");

}

}

**Output:**

Arithmetic Exception occurs

rest of the code

**5. Write a program to throw exception with your own message**

class InvalidAgeException extends Exception{

InvalidAgeException(String s){

super(s);

}

}

class TestCustomException1{

static void validate(int age)throws InvalidAgeException{

if(age<18)

throw new InvalidAgeException("not valid");

else

System.out.println("welcome to vote");

}

public static void main(String args[]){

try{

validate(13);

}catch(Exception m){System.out.println("Exception occured: "+m);}

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

}

}

**Output**:Exception occured: InvalidAgeException:not valid

rest of the code...

**6. Write a program to create your own exception**

class CustomException extends Exception {

String message;

CustomException(String str) {

message = str;

}

public String toString() {

return ("Custom Exception Occurred : " + message);

}

}

public class MainException {

public static void main(String args[]) {

try {

throw new CustomException("This is a custom message");

} catch(CustomException e) {

System.out.println(e);

}

}

}

**Output**

Custom Exception Occurred : This is a custom message

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

**Java finally example where exception doesn't occur**

public class TestFinallyBlock{

public static void main(String args[]){

try{

int data=25/5;

System.out.println(data);

}

catch(NullPointerException e){System.out.println(e);}

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

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

}

}

**Output**

$javac TestFinallyBlock.java

$java -Xmx128M -Xms16M TestFinallyBlock

5

finally block is always executed

rest of the code...

**Java finally example where exception occurs and not handled**

public class TestFinallyBlock1{

public static void main(String args[]){

try{

int data=25/0;

System.out.println(data);

}

catch(NullPointerException e){System.out.println(e);}

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

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

}

}

**Output**

$javac TestFinallyBlock1.java

$java -Xmx128M -Xms16M TestFinallyBlock1

finally block is always executed

Exception in thread "main" java.lang.ArithmeticException: / by zero

at TestFinallyBlock1.main(TestFinallyBlock1.java:4)

**Java finally example where exception occurs and handled**

public class TestFinallyBlock2{

public static void main(String args[]){

try{

int data=25/0;

System.out.println(data);

}

catch(ArithmeticException e){System.out.println(e);}

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

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

}

}

**Output**

$javac TestFinallyBlock2.java

$java -Xmx128M -Xms16M TestFinallyBlock2

java.lang.ArithmeticException: / by zero

finally block is always executed

rest of the 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);

}

}

**Output**

$javac ArithmeticExceptionTest.java

$java -Xmx128M -Xms16M ArithmeticExceptionTest

Exception in thread "main" java.lang.ArithmeticException: / by zero

at ArithmeticExceptionTest.main(ArithmeticExceptionTest.java:4)

**9. Write a program to generate ArrayIndexOutOfBoundException**

public class ArrayIndexOutOfBoundException {

public static void main(String[] args) {

String[] arr = {"Rohit","Shikar","Virat","Dhoni"};

//Declaring 4 elements in the String array

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

//Here, no element is present at the iteration number arr.length, i.e 4

System.out.println(arr[i]);

//So it will throw ArrayIndexOutOfBoundException at iteration 4

}

}

}

**Output**

$javac ArrayIndexOutOfBoundException.java

$java -Xmx128M -Xms16M ArrayIndexOutOfBoundException

Rohit

Shikar

Virat

Dhoni

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 4

at ArrayIndexOutOfBoundException.main(ArrayIndexOutOfBoundException.java:10)

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

public class Example {

public static void main(String args[]) {

try

{

Class.forName("GeeksForGeeks");

}

catch (ClassNotFoundException ex)

{

ex.printStackTrace();

}

}

}

**Output**

$javac Example.java

$java -Xmx128M -Xms16M Example

java.lang.ClassNotFoundException: GeeksForGeeks

at java.net.URLClassLoader.findClass(URLClassLoader.java:381)

at java.lang.ClassLoader.loadClass(ClassLoader.java:424)

at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:335)

at java.lang.ClassLoader.loadClass(ClassLoader.java:357)

at java.lang.Class.forName0(Native Method)

at java.lang.Class.forName(Class.java:264)

at Example.main(Example.java:6)

**11. Write a program to generate FileNotFoundException**

package org.arpit.java2blog;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

/\*\*

\* FileNotFoundException example

\* @author Arpit

\*

\*/

public class FileReadExample {

public static void main(String[] args) {

File file = new File("C:/Java2blog.txt");

FileInputStream fis = null;

try{

fis = new FileInputStream(file);

while (fis.read()!=-1){

System.out.println(fis.read());

}

}catch (FileNotFoundException e){

e.printStackTrace();

}catch (IOException e){

e.printStackTrace();

}finally{

try{

fis.close();

}catch (IOException e){

e.printStackTrace();

}

}

}

}

**Output**

$javac org/arpit/java2blog/FileReadExample.java

$java -Xmx128M -Xms16M org/arpit/java2blog/FileReadExample

java.io.FileNotFoundException: C:/Java2blog.txt (No such file or directory)

at java.io.FileInputStream.open0(Native Method)

at java.io.FileInputStream.open(FileInputStream.java:195)

at java.io.FileInputStream.<init>(FileInputStream.java:138)

at org.arpit.java2blog.FileReadExample.main(FileReadExample.java:16)

Exception in thread "main" java.lang.NullPointerException

at org.arpit.java2blog.FileReadExample.main(FileReadExample.java:26)

**12. Write a program to generate IOException**

import java.io.FileInputStream;

import java.io.FileNotFoundException;

public class FileNotFoundExceptionExample

{

public void checkFileNotFound()

{

try

{

FileInputStream in = new FileInputStream("input.txt");

System.out.println("This is not printed");

}

catch (FileNotFoundException fileNotFoundException)

{

fileNotFoundException.printStackTrace();

}

}

public static void main(String[] args)

{

FileNotFoundExceptionExample example = new FileNotFoundExceptionExample();

example.checkFileNotFound();

}

}

**Output**

$javac FileNotFoundExceptionExample.java

$java -Xmx128M -Xms16M FileNotFoundExceptionExample

java.io.FileNotFoundException: input.txt (No such file or directory)

at java.io.FileInputStream.open0(Native Method)

at java.io.FileInputStream.open(FileInputStream.java:195)

at java.io.FileInputStream.<init>(FileInputStream.java:138)

at java.io.FileInputStream.<init>(FileInputStream.java:93)

at FileNotFoundExceptionExample.checkFileNotFound(FileNotFoundExceptionExample.java:9)

at FileNotFoundExceptionExample.main(FileNotFoundExceptionExample.java:20)

**13. Write a program to generate NoSuchMethodException**

private static Method getEquals() {

if (equalsMth == null) {

try {

equalsMth = Object.class.getMethod("equals", Object.class); // NOI18N

} catch (NoSuchMethodException e) {

e.printStackTrace();

}

}

return equalsMth;

}

**14. Write a program to generate NullPointerException**

// To use randomUUID function.

import java.util.UUID;

import java.io.\*;

class Singleton

{

// Initializing values of single and ID to null.

private static Singleton single = null;

private String ID = null;

private Singleton()

{

/\* Make it private, in order to prevent the

creation of new instances of the Singleton

class. \*/

// Create a random ID

ID = UUID.randomUUID().toString();

}

public static Singleton getInstance()

{

if (single == null)

single = new Singleton();

return single;

}

public String getID()

{

return this.ID;

}

}

// Driver Code

public class TestSingleton

{

public static void main(String[] args)

{

Singleton s = Singleton.getInstance();

System.out.println(s.getID());

}

}

**Output**

$javac TestSingleton.java

$java -Xmx128M -Xms16M TestSingleton

f15581a8-7ced-4311-8d8d-36b975ede495

**15. Write a program to generate NumberFormatException**

public class Example {

public static void main(String[] args) {

int a = Integer.parseInt(null); //throws Exception as //the input string is of illegal format for parsing as it is null.

}

}

**Output**

$javac Example.java

$java -Xmx128M -Xms16M Example

Exception in thread "main" java.lang.NumberFormatException: null

at java.lang.Integer.parseInt(Integer.java:542)

at java.lang.Integer.parseInt(Integer.java:615)

at Example.main(Example.java:4)

**16. Write a program to generate StringIndexOutOfBoundsException**

public class StringIndexOutOfBoundsException {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the input text : ");

try{

char c = scanner.nextLine().charAt(4);

System.out.println("The character at index 4 is : "+c);

}catch(Exception e){

//StringIndexOutOfBoundsException cannot be caught explicitly

System.out.println("Caught exception : "+e.toString());

}

}

}

**17. Write a program to generate SQLException**

public static void commit() {

Connection conn = tl.get();

if (conn != null) {

try {

conn.commit();

} catch (SQLException e) {

e.printStackTrace();

throw new RuntimeException("Connection 事务提交异常");

}

}

}