**Exception Handling**

Exception – abnormal termination.

It interrupt the normal flow of execution

Exception Handling –

It avoid the abnormal termination.

Exception arise in a code at run time or Exception is a run time error

Java introduced five keywords to handle the exception

try - is use to monitor the code.

catch - is used to handle the exception which is thrown by try block.

throw – manually throw and exception

throw mostly used for customized exception

throws – exception is thrown out of a method.

finally

**Example :**

**public class Car {**

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

**// TODO Auto-generated method stub**

**System.*out*.println("one");**

**System.*out*.println("two");**

**System.*out*.println("three");**

**try {**

**System.*out*.println(10/0); // ArithmeticException**

**}catch(ArithmeticException e) {**

**System.*out*.println(e);**

**}**

**System.*out*.println("four");**

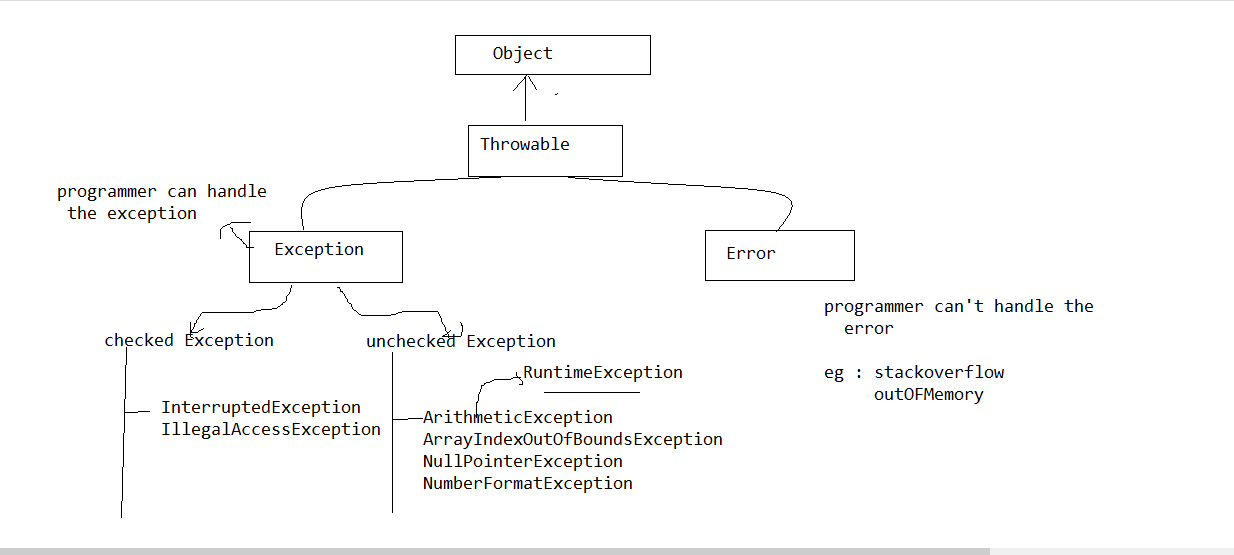
**System.*out*.println("five");**

**}**

**}**

**Exception Types**

All exception Type super class is Throwable ( class),



**Multiple catch Clause :**

More than one exception could be raise by single piece of code.

try {

}catch( ) {

}catch() {

}

Example 1 :

**public** **class** MultiCatch {

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

// **TODO** Auto-generated method stub

**try** {

**int** a=5/1; // ArithmeticException

**int** c[]= {6};

c[1]=15; // ArrayIndexOutOfBoundsException

}**catch**(ArithmeticException e) {

System.***out***.println(e);

}**catch**(ArrayIndexOutOfBoundsException e) {

System.***out***.println(e);

}

}

}

Example 2 :

**public class MultiCatch1 {**

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

**// TODO Auto-generated method stub**

**try {**

**int a=5/1; // ArithmeticException**

**int c[]= {6};**

**c[1]=15; // ArrayIndexOutOfBoundsException**

**}catch(ArithmeticException e) {**

**System.*out*.println(e);**

**}catch(Exception e) {**

**System.*out*.println(e);**

**}**

**}**

**}**

**Nested try :**

Inside the try there is an another try

try {

try {

}catch( ) {

}

}

catch( ) {

}

**public class NestedTry {**

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

**// TODO Auto-generated method stub**

**try {**

**int a=5/1; // ArithmeticException**

**try {**

**int c[]= {4};**

**c[1]=6; // ArrayIndexOutOfBoundsException**

**}catch(NullPointerException e) {**

**System.*out*.println(e);**

**}**

**}catch(ArithmeticException e) {**

**System.*out*.println(e);**

**}catch(ArrayIndexOutOfBoundsException e) {**

**System.*out*.println(e);**

**}**

**}**

**}**

**Throw**

Example program

**public class ThrowDemo {**

**static void validate(int age) {**

**if(age<20) {**

**throw new NullPointerException("not eligible to vote");**

**}**

**}**

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

**// TODO Auto-generated method stub**

**try {**

***validate*(10);**

**}catch(NullPointerException e) {**

**System.*out*.println(e);**

**}**

**}**

**}**

Throws

void father() {

child()

}

void child() {

// mistake

}

**Example Program :**