**Try Catch in Java – Exception handling**

**What is Try Block?**

The try block contains a block of program statements within which an exception might occur. A try block is always followed by a catch block, which handles the exception that occurs in associated try block. A try block must followed by a Catch block or Finally block or both.

**Syntax of try block**

try{

//statements that may cause an exception

}

**What is Catch Block?**

A catch block must be associated with a try block. The corresponding catch block executes if an exception of a particular type occurs within the try block. For example if an [**arithmetic exception**](http://beginnersbook.com/2013/04/exception-handling-examples/) occurs in try block then the statements enclosed in catch block for arithmetic exception executes.

**Syntax of try catch in java**

try

{

//statements that may cause an exception

}

catch (exception(type) e(object))‏

{

//error handling code

}

**Flow of try catch block**

1. If an exception occurs in try block then the control of execution is passed to the catch block from try block. The exception is caught up by the corresponding catch block. A single try block can have multiple catch statements associated with it, but each catch block can be defined for only one exception class. The program can also contain [**nested**](http://beginnersbook.com/2013/04/nested-try-catch/) [**try-catch-finally blocks**](http://beginnersbook.com/2013/05/flow-in-try-catch-finally/).
2. After the execution of all the try blocks, the code inside the finally block executes. It is not mandatory to include a finally [**block**](http://beginnersbook.com/2013/04/java-finally-block/) at all, but if you do, it will run regardless of whether an exception was thrown and handled by the try and catch blocks.

**An example of Try catch in Java**

class Example1 {

public static void main(String args[]) {

int num1, num2;

try {

// Try block to handle code that may cause exception

num1 = 0;

num2 = 62 / num1;

System.out.println("Try block message");

} catch (ArithmeticException e) {

// This block is to catch divide-by-zero error

System.out.println("Error: Don't divide a number by zero");

}

System.out.println("I'm out of try-catch block in Java.");

}

}

Output:

Error: Don't divide a number by zero

I'm out of try-catch block in Java.

**Multiple catch blocks in Java**

1. A try block can have any number of catch blocks.  
2. A catch block that is written for catching the class Exception can catch all other exceptions  
Syntax:

catch(Exception e){

  //This catch block catches all the exceptions

}

3. If multiple catch blocks are present in a program then the above mentioned catch block should be placed at the last as per the exception handling best practices.  
4. If the try block is not [**throwing any exception**](http://beginnersbook.com/2013/04/throw-in-java/), the catch block will be completely ignored and the program continues.  
5. If the try block [**throws an exception**](http://beginnersbook.com/2013/04/java-throws/), the appropriate catch block (if one exists) will catch it  
–catch(ArithmeticException e) is a catch block that can catch ArithmeticException  
–catch(NullPointerException e) is a catch block that can catch NullPointerException  
6. All the statements in the catch block will be executed and then the program continues.

**Example of Multiple catch blocks**

class Example2{

public static void main(String args[]){

try{

int a[]=new int[7];

a[4]=30/0;

System.out.println("First print statement in try block");

}

catch(ArithmeticException e){

System.out.println("Warning: ArithmeticException");

}

catch(ArrayIndexOutOfBoundsException e){

System.out.println("Warning: ArrayIndexOutOfBoundsException");

}

catch(Exception e){

System.out.println("Warning: Some Other exception");

}

System.out.println("Out of try-catch block...");

}

}

Output:

Warning: ArithmeticException

Out of try-catch block...

In the above example there are multiple catch blocks and these catch blocks executes sequentially when an exception occurs in try block. Which means if you put the last catch block ( catch(Exception e)) at the first place, just after try block then in case of any exception this block will execute as it has the [**ability to handle all exceptions**](http://beginnersbook.com/2013/04/java-exception-handling/). This catch block should be placed at the last to avoid such situations.