Try, catch, throw and throws in Java

Difficulty Level : Easy Last Updated : 06 Feb, 2019

What is an Exception?

An exception is an “unwanted or unexpected event”, which occurs during the execution of the program i.e, at run-time, that disrupts the normal flow of the program’s instructions. When an exception occurs, execution of the program gets terminated.

Why does an Exception occurs?

An exception can occur due to several reasons like Network connection problem, Bad input provided by user, Opening a non-existing file in your program etc

Blocks & Keywords used for exception handling

1.try: The try block contains set of statements where an exception can occur.

```
try
{
    // statement(s) that might cause exception
}
```

2.**catch** : Catch block is used to handle the uncertain condition of try block. A try block is always followed by a catch block, which handles the exception that occurs in associated try block.

```
catch
{
   // statement(s) that handle an exception
   // examples, closing a connection, closing
   // file, exiting the process after writing
   // details to a log file.
}
```

3.**throw**: Throw keyword is used to transfer control from try block to catch block.

4.**throws**: Throws keyword is used for exception handling without try & catch block. It specifies the exceptions that a method can throw to the caller and does not handle itself.

5.**finally**: It is executed after catch block. We basically use it to put some common code when there are multiple catch blocks.

Example of an Exception generated by system is given below :

```
Exception in thread "main"
java.lang.ArithmeticException: divide
by zero at ExceptionDemo.main(ExceptionDemo.java:5)
ExceptionDemo: The class name
main:The method name
ExceptionDemo.java:The file name
java:5:line number
```

```java
// Java program to demonstrate working of try,
// catch and finally

class Division {
    public static void main(String[] args)
    {
        int a = 10, b = 5, c = 5, result;
        try {
            result = a / (b - c);
            System.out.println("result" + result);
        }

        catch (ArithmeticException e) {
            System.out.println("Exception caught:Division by zero");
        }

        finally {
            System.out.println("I am in final block");
        }
    }
}
```

Output:

```
Exception caught:Division by zero
I am in final block
```

An example of throws keyword:

```java
// Java program to demonstrate working of throws
```

```
class ThrowsExecp {

    // This method throws an exception
    // to be handled
    // by caller or caller
    // of caller and so on.
    static void fun() throws IllegalAccessException
    {
        System.out.println("Inside fun(). ");
        throw new IllegalAccessException("demo");
    }

    // This is a caller function
    public static void main(String args[])
    {
        try {
            fun();
        }
        catch (IllegalAccessException e) {
            System.out.println("caught in main.");
        }
    }
}
```

Output:

```
Inside fun().
caught in main.
```