

Exception Handling Rules of Thumb - by Robert Bushman

Good exception handling is an important, though often overlooked aspect of software development. A small investment in better exception handling will make most software more robust and responsive. The following rules all have exceptions, but in most cases they will serve you well.

Be Suspicious of Long Try Blocks

Shoot for one statement per try block. This will allow a more precise catch block, tailored to the exact fault in question.

If "This Can Never Happen," Throw an Error

The one line comment "This can never happen" is the perfect place to throw an error. Errors are unchecked and it will never happen, so it costs nothing.

Catch The Specific Exception Type

Catch the specific type that may be thrown. The intent of the catch block will be more clear, and you won't accidentally catch some other exception.

If You Catch Throwable or Error, Do So at The Top

Errors typically mean something serious went wrong. Catch specific errors if they are recoverable. Catch the generic Throwable or Error near the entry point of the program, if at all.

Throw Specific Exception Types

Throwing a generic Exception does not communicate the nature of the problem. If there is not a specific exception type for the condition in question, write a new one.

Use Meaningful Exception Messages

Clearly state what went wrong and how the invoker can avoid the problem in the future. Don't be stingy; if it takes fifty words to explain the problem, use fifty words.

Use Chained Exceptions

If you are catching one type and throwing another, chain the cause by passing the original exception to the constructor of the new Exception.

Create Your Own Exceptions

In some cases you may want to return some info beyond a simple message along with the exception. Don't be afraid to write a new exception type that has appropriate parameters.

Using Checked Exceptions

Checked exceptions are part of the API of the method - they should be taken as seriously as the return type. They should be clear, meaningful, and specific to the problem at hand.