## **Lesson: Exceptions**

# What Is an Exception?

An *exception* is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions. When an error occurs within a method, the method creates an object and hands it off to the runtime system. The object, called an *exception object*, contains information about the error, including its type and the state of the program when the error occurred. After a method throws an exception, the runtime system attempts to find something to handle it.

# The Catch or Specify Requirement

The code that might throw certain exceptions must be enclosed by either of the following:

* A *try* statement that catches the exception. The *try* must provide a handler for the exception, as described in [Catching and Handling Exceptions](http://docs.oracle.com/javase/tutorial/essential/exceptions/handling.html).
* A method that specifies that it can throw the exception. The method must provide a *throws* clause that lists the exception, as described in [Specifying the Exceptions Thrown by a Method](http://docs.oracle.com/javase/tutorial/essential/exceptions/declaring.html).

Code that fails to honor the Catch or Specify Requirement will not compile.

Not all exceptions are subject to the Catch or Specify Requirement. Exist three basic categories of exceptions, only one of which is subject to the Requirement:

The first kind of exception is the checked exception. These are exceptional conditions that a well-written application should anticipate and recover from. A well-written program will catch this exception and notify the user of the mistake. Checked exceptions are subject to the Catch or Specify Requirement.

The second kind of exception is the error. These are exceptional conditions that are external to the application, and that the application usually cannot anticipate or recover from. An application might choose to catch this exception, in order to notify the user of the problem — but it also might make sense for the program to print a stack trace and exit. Errors are not subject to the Catch or Specify Requirement.

The third kind of exception is the runtime exception. These are exceptional conditions that are internal to the application, and that the application usually cannot anticipate or recover from. These usually indicate programming bugs, such as logic errors or improper use of an API. The application can catch this exception, but it probably makes more sense to eliminate the bug that caused the exception to occur. Runtime exceptions are not subject to the Catch or Specify Requirement.