Exception Handling

An exception (or exceptional event) is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program/Application terminates abnormally, which is not recommended, therefore, these exceptions are to be handled.

* An exception can occur for many different reasons. Following are some scenarios where an exception occurs.
* A user has entered an invalid data.
* A file that needs to be opened cannot be found.
* A network connection has been lost in the middle of communications or the JVM has run out of memory.

Methods:

1. public String getMessage(): Returns a detailed message about the exception that has occurred. This message is initialized in the Throwable constructor.
2. public Throwable getCause(): Returns the cause of the exception as represented by a Throwable object.
3. public String toString(): Returns the name of the class concatenated with the result of getMessage().
4. public void printStackTrace(): Prints the result of toString() along with the stack trace to System.err, the error output stream.
5. public StackTraceElement [] getStackTrace(): Returns an array containing each element on the stack trace. The element at index 0 represents the top of the call stack, and the last element in the array represents the method at the bottom of the call stack.
6. public Throwable fillInStackTrace(): Fills the stack trace of this Throwable object with the current stack trace, adding to any previous information in the stack trace.