Discussion Questions for "Throwing Exceptions"

1. Can you think of Java code you have written where throwing an exception would be useful? Give an example or two.

2. In Java, there is an "assert" statement that allows you to specify a Boolean expression, and if that expression is true, nothing happens. If the expression is false, java throws an assertion error. It is possible at run time to run with or without assertions by using a parameter to the "java" command. Where might you use an assert statement instead of throwing an exception yourself? Why?

3. What is the audience that would understand the messages generated by an uncaught exception? Put yourself in the place of someone not familiar with Java who sees the result of an uncaught exception. Would that person know how to react to the messages?

4. Why is it important to print out the entire program stack, rather than just the specific line of code that generated an exception?

5. Java makes a distinction between the "Throwable" class and the "Exception" class, which is a sub-class of Throwable. Can you think of any examples where you might want to make something that is throwable, but is not an exception?