Chapter 5: Summary

Summary

Here's a matching game to help you recall some of the terms we covered in this lesson.

Text equivalent start.

Instructions: Read the clue in the first column, and guess the term. Then read the second column for the answer.

Clue	**Answer**
Unlikely to crash.	Robust
A keyword that tells Java to run a class without creating an object first.	Static
Self-contained.	Encapsulated
An event that interrupts the normal flow of a program's logic.	Exception
The act of trapping and processing an exception.	Catching
Code that the program should execute whether there's an exception or not.	Finally block
A keyword that lets an upcoming class inherit attributes and methods from another class	Extends

Text equivalent stop.

I hope you did well on the game. If you had trouble, please post a message in the Discussion Area telling me which parts of the lesson you didn't understand.

Error handling is always an issue in computer programs. The more robust a program is, the less likely it is to suffer a catastrophic failure. We've seen a couple of ways to improve our classes in this lesson:

- **Static methods:** A user can call these to check on values before trying to put values into an object.
- **Throwing checked exceptions:** We can do this when we run into a situation that just doesn't work. Throwing checked exceptions also forces a user to handle them when calling the methods, since the compiler makes sure to handle them somewhere in the call stack.

Be sure to work through the assignment, and let me know in the Discussion Area if you have any questions about it or about the lesson.

We're finally done with our temperature conversion example; I think we've just about beaten it to death, don't you? Next time we'll look at a couple of other Java features: its switch statement and enumerations. Those are different ways to check values and to branch to alternative actions based on the values.