Return to Module Overview Page (https://onlinelearning.berkeley.edu/courses/665040/pages/module-ten)

10.2: The Reflection API

Occasionally, it is necessary to learn more about an object at run time than you know at compile time. This most often comes up in development tools; for example, a GUI builder that is written to deal with generic component objects will need to learn more about individual components to determine the best way to arrange them. Another use of this runtime discovery capability would be in an object-oriented database browser; a database might be defined to hold references to java.lang.Object (the base class of all Java objects, so the database could actually hold any object). On the other hand, to display the contents of the database, a viewing tool would have to interrogate each object to find out its real type, and then optionally call some of each object's methods to display its contents.

Early in this course, I warned you that using instanceof to test for an object's type, then acting accordingly, was a Bad Thing. Now that you've traveled this long road, however, it's OK to tell you that sometimes you have no other choice, especially if the classes you're testing for were not all designed together, and perhaps don't share a common inheritance tree.

The Java Reflection API lets you discover everything there is to know about an object at runtime. Here I will present just a few examples of its use. You may never need to use it, but if you do, you will be glad it's there!