

## Class Hierarchy - Update

Thus far, we have worked with the `String` class, the `Math` class, the `StringTokenizer`, and Java's eight wrapper classes. Let's pause briefly, and re-visit a topic from early: class hierarchy. We noted that the `Object` class lies at the top of Java's class hierarchy. All classes, either directly or indirectly (through other classes), inherit characteristics from the `Object` class. Earlier, we illustrated the superclass-subclass relationship between the `String` and `Object` classes. Figure 1 expands on this, showing the class hierarchy of all the classes presented thus far.

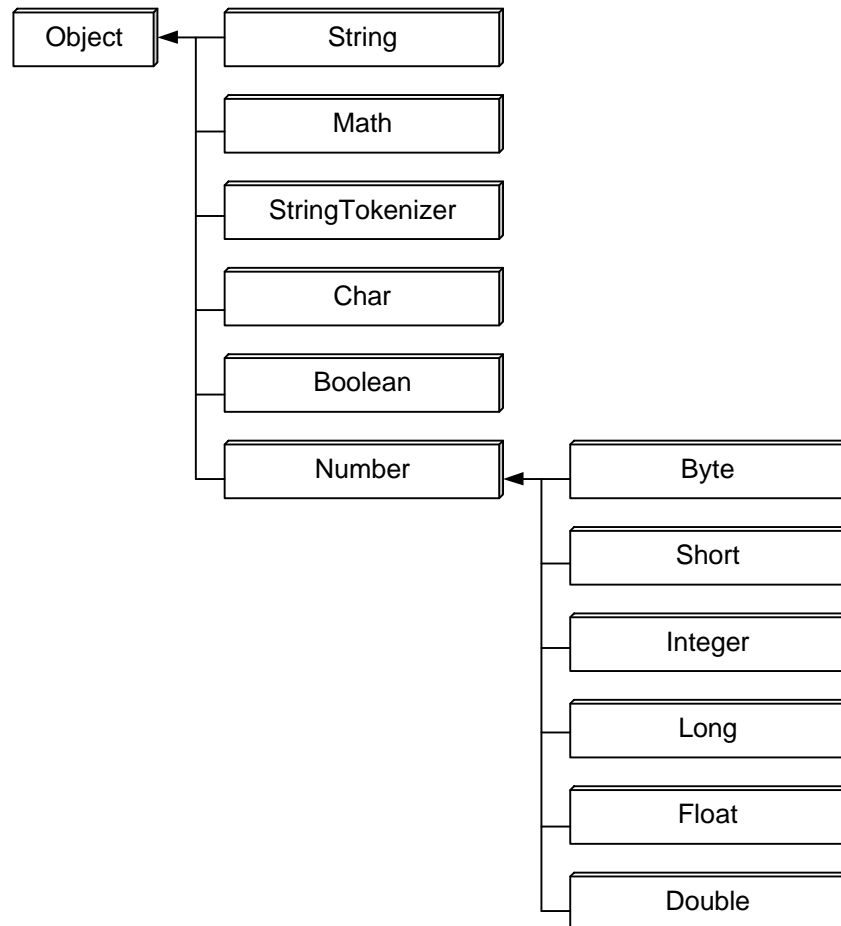


Figure 1. Class hierarchy update

As you might imagine, a complete class hierarchy diagram for all the classes in the Java API could fill a large wall-size poster. (In fact, such posters are available from Java vendors.) A complete diagram would not only be "taller" than Figure 1, it would also be "wider" because many classes are "subclasses of subclasses", and so on. We see an example of this in the six numeric wrapper classes in Figure 1. Each is a subclass of the `Number` class, which, in turn, is a subclass of the `Object` class. We will present portions of the class hierarchy on an as-needed basis as we continue our travels through the Java API.