4.14 Primitive Types

The table in Appendix D lists the eight primitive types in Java. Like C and C++, Java requires all variables to have a type.2

2. We'll see an exception to this with lambdas in Chapter 17, Lambdas and Streams.

In C and C++, programmers frequently have to write separate versions of programs to support different computer platforms, because the primitive types are not guaranteed to be identical from computer to computer. For example, an `int` on one machine might be represented by 16 bits (2 bytes) of memory, on a second machine by 32 bits (4 bytes), and on another machine by 64 bits (8 bytes). In Java, `int` values are always 32 bits (4 bytes).

Portability Tip 4.1

The primitive types in Java are portable across all computer platforms that support Java.

Each type in Appendix D is listed with its size in bits (there are eight bits to a byte) and its range of values. Because the designers of Java want to ensure portability, they use

internationally recognized standards for character formats (Unicode; for more information, visit `http://www.unicode.org`) and floating-point numbers (IEEE 754; for more information, visit `http://grouper.ieee.org/groups/754/`).

Recall from Section 3.2 that variables of primitive types declared outside of a method as instance variables of a class are *automatically assigned default values unless explicitly initialized*. Instance variables of types `char`, `byte`, `short`, `int`, `long`, `float` and `double` are all given the value `0` by default. Instance variables of type `boolean` are given the value `false` by default. Reference-type instance variables are initialized by default to the value `null`.