14.2 Fundamentals of Characters and Strings

Characters are the fundamental building blocks of Java source programs. Every program is composed of a sequence of characters that—when grouped together meaningfully—are interpreted by the Java compiler as a series of instructions used to accomplish a task. A program may contain **character literals**. A character literal is an integer value represented as a character in single quotes. For example, `'z'` represents the integer value of `z`, and `'\t'` represents the integer value of a tab character. The value of a character literal is the integer value of the character in the **Unicode character set**. Appendix B presents the integer equivalents of the characters in the ASCII character set, which is a subset of Unicode (discussed in online Appendix H).

Recall from Section 2.2 that a string is a sequence of characters treated as a single unit. A string may include letters, digits and various **special characters**, such as `+`, `-`, `*`, `/` and `$`. A string is an object of class `String`. **String literals** (stored in memory as `String` objects) are written as a sequence of characters in double quotation marks, as in:

`"John Q. Doe"`	(a name)
`"9999 Main Street"`	(a street address)

`"Waltham, Massachusetts"`	(a city and state)
`"(201) 555-1212"`	(a telephone number)

A string literal may be assigned to a `String` reference. The declaration

```
String color = "blue";
```

initializes `String` variable `color` to refer to a `String` object that contains the string `"blue"`.

Performance Tip 14.1

To conserve memory, Java treats all string literals with the same contents as a single `String` *object that has many references to it.*