1.1 Getting Started

Building data structures and algorithms requires that we communicate detailed instructions to a computer. An excellent way to perform such communication is using a high-level computer language, such as Java. In this chapter, we provide an overview of the Java programming language, and we continue this discussion in the next chapter, focusing on object-oriented design principles. We assume that readers are somewhat familiar with an existing high-level language, although not necessarily Java. This book does not provide a complete description of the Java language (there are numerous language references for that purpose), but it does introduce all aspects of the language that are used in code fragments later in this book.

We begin our Java primer with a program that prints "Hello Universe!" on the screen, which is shown in a dissected form in Figure 1.1.

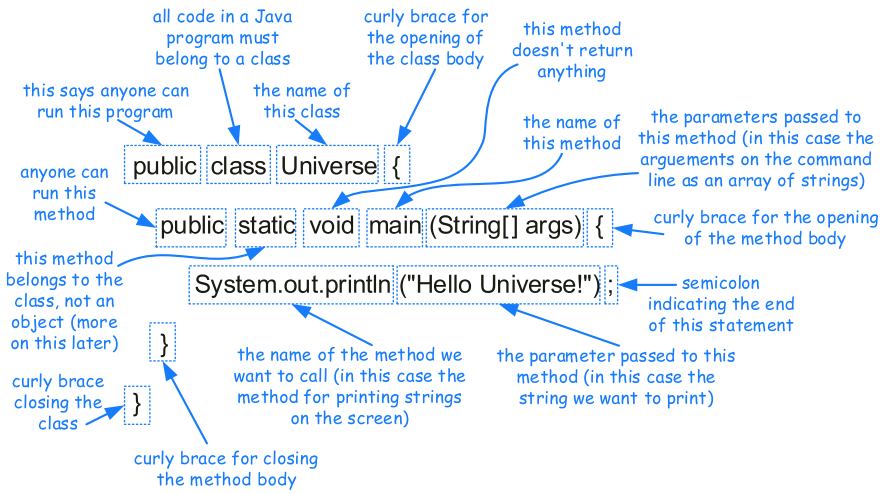

Figure 1.1: A "Hello Universe!" program.

In Java, executable statements are placed in functions, known as ***methods***, that belong to ***class*** definitions. The Universe class, in our first example, is extremely simple; its only method is a static one named main, which is the first method to be executed when running a Java program. Any set of statements between the braces "{" and "}" define a program ***block***. Notice that the entire Universe class definition is delimited by such braces, as is the body of the main method.

The name of a class, method, or variable in Java is called an ***identifier***, which can be any string of characters as long as it begins with a letter and consists of letters, numbers, and underscore characters (where "letter" and "number" can be from any written language defined in the Unicode character set). We list the exceptions to this general rule for Java identifiers in Table 1.1.