# 1 Running Java: Compilation, Loading, and Execution

Before a Java program can be executed, it must be compiled and loaded. The compiler checks that the Java program is *legal:* that the program conforms to the Java syntax (grammar), that operators (such as +) are applied operands (such as 5 and x) of the correct type, and so on. If so, the compiler generates so-called *class files*. Execution then starts by loading the needed class files.

Thus running a Java program involves three stages: *compilation* (checks that the program is well-formed), *loading* (loads and initializes classes), and *execution* (runs the program code).

# 2 Names and Reserved Names

A *legal name* (of a variable, method, field, parameter, class, interface or package) starts with a letter or dollar sign ($) or underscore (_), and continues with zero or more letters or dollar signs or underscores or digits (0–9). Avoid dollar signs in class names. Uppercase letters and lowercase letters are considered distinct. A legal name cannot be one of the following *reserved names:*

```
abstract   char       else       for          interface  protected  switch        try
assert     class      enum       goto         long       public     synchronized  void
boolean    const      extends    if           native     return     this          volatile
break      continue   false      implements   new        short      throw         while
byte       default    final      import       null       static     throws
case       do         finally    instanceof   package    strictfp   transient
catch      double     float      int          private    super      true
```

# 3 Java Naming Conventions

The following naming conventions are often followed, although not enforced by Java:

- If a name is composed of several words, then each word (except possibly the first one) begins with an uppercase letter. Examples: `setLayout`, `addLayoutComponent`.

- Names of variables, fields, and methods begin with a lowercase letter. Examples: `vehicle`, `myVehicle`.

- Names of classes and interfaces begin with an uppercase letter. Examples: `Cube`, `ColorCube`.

- Named constants (such as `final static` fields) are written entirely in uppercase, and the parts of composite names are separated by underscores (_). Examples: `CENTER`, `MAX_VALUE`.

- Package names are sequences of dot-separated lowercase names. Example: `java.awt.event`. For uniqueness, they are often prefixed with reverse domain names, as in `com.sun.xml.util`.

# 4 Comments and Program Layout

*Comments* have no effect on the execution of the program but may be inserted anywhere to help humans understand the program. There are two forms: one-line comments and delimited comments.

*Program layout* has no effect on the computer's execution of the program but is used to help humans understand the structure of the program.