Supplement: Using Packages

For Introduction to Java Programming
By Y. Daniel Liang

NOTE: If you wish to use packages to organize the classes in the text, read this supplement after Chapter 1 in the text. Assume that the classes in Chapter*i* will be placed in package chapter*i*.

Packages can be used to organize classes. To do so, you need to add the following line as the first noncomment and nonblank statement in the program:

```
package packagename;
```

Listing 1 gives a program that places class **Welcome** in package **chapter1**.

Listing 1 Welcome.java

```
/** Use package for the class */
package chapter1;

public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Listing 1 is identical to Listing 1.1 in the text except that the **Welcome** class in Listing 1 is placed in package **chapter1**. A package corresponds to a directory. You need to create a directory named **chapter1** and place Welcome.java in the directory. If you use an IDE such as NetBeans and Eclipse, the directory is automatically created. Suppose all source code in chapter*i* are placed in the directory **chapter*i*** in this text, as shown in Figure 1.

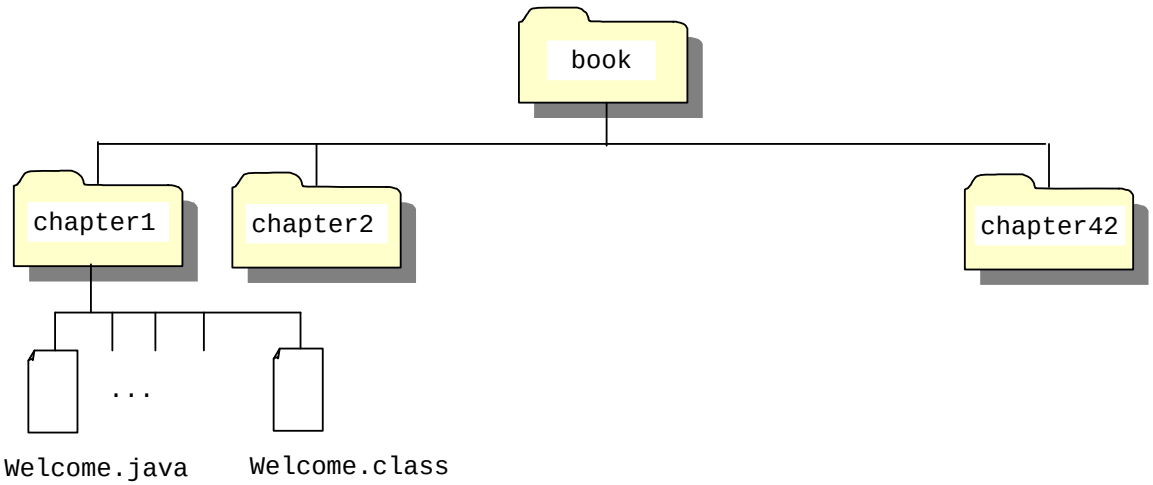

Figure 1

The .java and .class files in this book are placed in packages.

NOTE
The root directory where the .class files (including the packages) are stored is known as the *classpath* directory. In this book, our classpath is **c:\book**.

NOTE
To compile and run programs from the command window rather using an IDE, you need to know at least two DOS commands: **mkdir** and **cd**.

- **mkdir dirName** -- Creates a new directory named dirName.
- **cd dirName** -- Changes to the specified directory. For example, **cd c:\book** changes to the directory c:\book.
- **cd ..** -- Changes to the parent directory.

See Supplement I.C, "Creating, Compiling and Running Java Programs from the Command Window," for other useful commands.

To compile Welcome.java from the command window, change the directory to **chapter1**, and type **javac Welcome.java**.
To run the class, change to the classpath directory, and type **java chapter1.Welcome**, as shown in Figure 2.

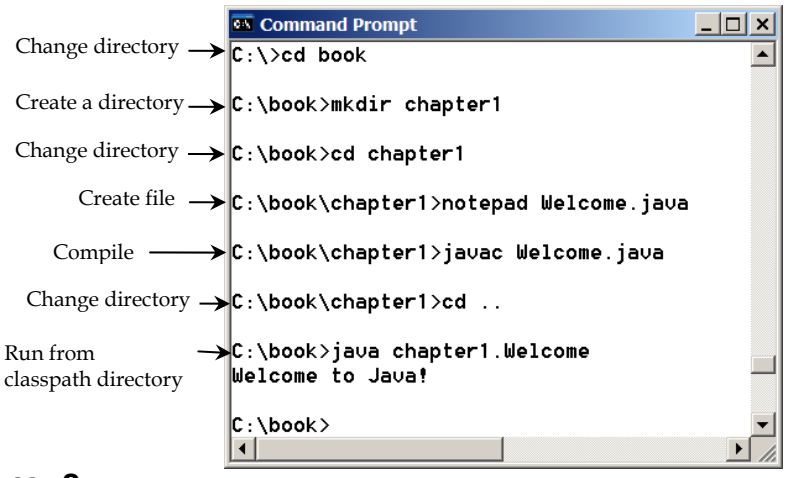

Figure 2
You must run a class from the classpath directory.

NOTE

If a class is defined without the package statement, the class is said to be placed in the *default package*. The **Welcome** class in Listing 1.1 in the text is placed in the default package.