Supplement I.F: 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 Section 1.7 in the text. Assume that the classes in Chapter*i* will be placed in package chapter*i*.

1.11 Placing Classes in a Package

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 1.2 gives a program that places class Welcome in package chapter1.

Listing 1.2 Welcome.java

******PD: Please add line numbers (including space lines, true for all line numbers in the book) in the following code******
******Layout: Please layout exactly. Don’t skip the space. This is true for all source code in the book. Thanks, AU.***
<Side Remark line 1: paragraph comment>
<Side Remark line 2: package>
<Side Remark line 4: main method>
<Side Remark line 6: display message>

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

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

Listing 1.2 is identical to Listing 1.1 except that the Welcome class in Listing 1.2 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, Eclipse, or JBuilder, the directory is automatically created. From now on, all source code in chapter*i* are placed in the directory chapter*i* in this text, as shown in Figure 1.14.

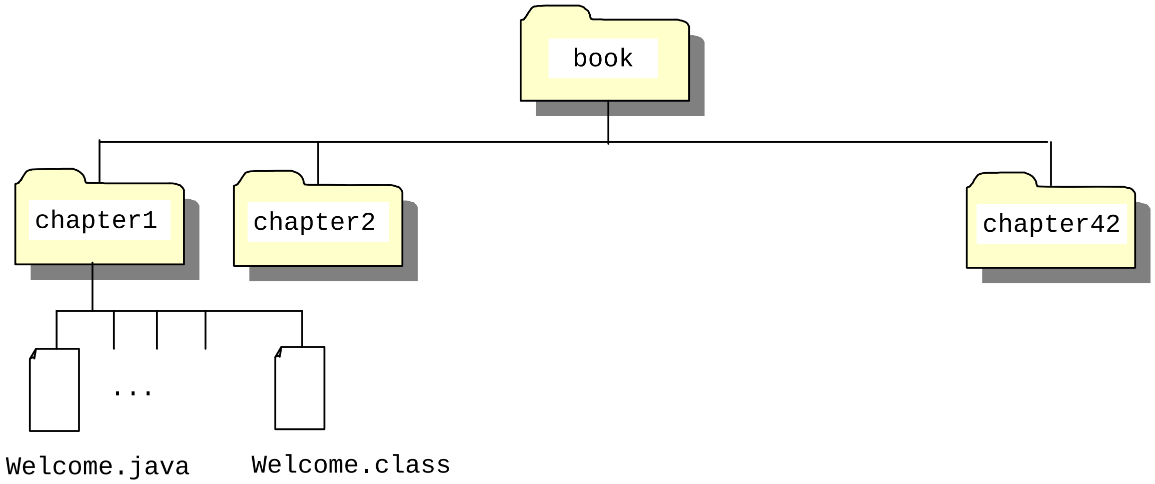


Figure 1.14

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

NOTE

<Side Remark: classpath>

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**.

****End of NOTE***

NOTE

<Side Remark: DOS commands>

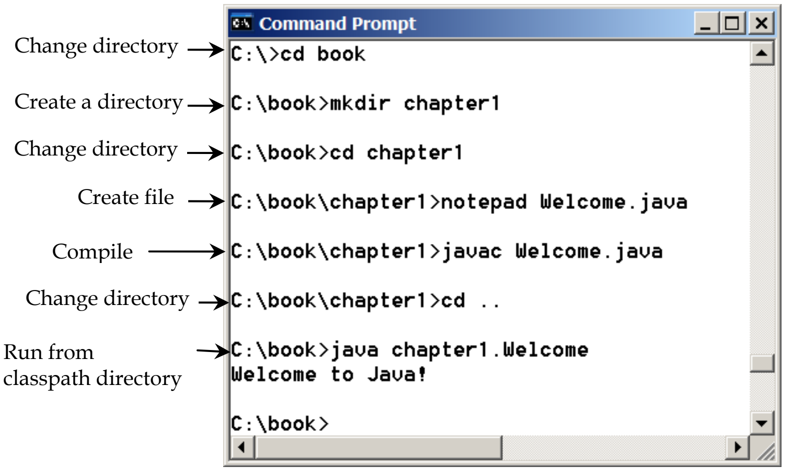
To compile and run programs from the command window rather using an IDE, you need to know at least two DOS commands: **mkdir** and **cd**.

- **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.

End of NOTE

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 1.15.



```
C:\>cd book

C:\book>mkdir chapter1

C:\book>cd chapter1

C:\book\chapter1>notepad Welcome.java

C:\book\chapter1>javac Welcome.java

C:\book\chapter1>cd ..

C:\book>java chapter1.Welcome
Welcome to Java!

C:\book>
```

Figure 1.15
You must run a class from the classpath directory.

NOTE

<Side Remark: default package>

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