16.1 Reading and Writing Text Files

We begin this chapter by discussing the common task of reading and writing files that contain text. Examples are files that are created with a simple text editor, such as Windows Notepad, as well as Java source code and HTML files.

The simplest mechanism for reading text is to use the Scanner class. You already know how to use a Scanner for reading console input. To read input from a disk file, first construct a FileReader object with the name of the input file, then use the FileReader to construct a Scanner object:

```java
FileReader reader = new FileReader("input.txt");
Scanner in = new Scanner(reader);
```

This Scanner object reads text from the file input.txt. You can use the Scanner methods (such as next, nextLine, nextInt, and nextDouble) to read data from the input file.

To write output to a file, you construct a PrintWriter object with the given file name, for example:

```java
PrintWriter out = new PrintWriter("output.txt");
```

If the output file already exists, it is emptied before the new data are written into it. If the file doesn't exist, an empty file is created.

Use the familiar print and println methods to send numbers, objects, and strings to a PrintWriter:

```java
out.println(29.95);
out.println(new Rectangle(5, 10, 15, 25));
out.println("Hello, World!");
```

The print and println methods convert numbers to their decimal string representations and use the toString method to convert objects to strings.

When you are done writing to a PrintWriter, be sure to close it:

```java
out.close();
```

If your program exits without closing the PrintWriter, not all of the output may be written to the disk file.
16.1  Reading and Writing Text Files

The following program puts these concepts to work. It reads all lines of an input file and sends them to the output file, preceded by line numbers. If the input file is

Mary had a little lamb
Whose fleece was white as snow.
And everywhere that Mary went,
The lamb was sure to go!

then the program produces the output file

/* 1 */ Mary had a little lamb
/* 2 */ Whose fleece was white as snow.
/* 3 */ And everywhere that Mary went,
/* 4 */ The lamb was sure to go!

The line numbers are enclosed in /* */ delimiters so that the program can be used for numbering Java source files.

File LineNumberer.java

```java
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;

public class LineNumberer {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        System.out.println("Input file: ");
        String inputFileName = console.nextLine();
        System.out.println("Output file: ");
        String outputFileName = console.nextLine();

        try {
            FileReader reader = new FileReader(inputFileName);
            Scanner in = new Scanner(reader);
            PrintWriter out = new PrintWriter(outputFileName);
            int lineNumber = 1;

            while (in.hasNextLine())
                {
                    String line = in.nextLine();
                    out.println("/* " + lineNumber + " */ + line);
                    lineNumber++;
                }
            out.close();
        }
        catch (IOException exception)
            {
                System.out.println("Error processing file: " + exception);
            }
    }
```