Frank Roetker CM2419 CSSE221-01
Eric Guilford CM2221 Dr. Defoe

September 27, 2011

Capsule Project: File IO

Source: Horstmann

What is File IO and when should I use it?

Sometimes you want to incorporate file input and output in your project. Using File IO in your software is great when you want to keep data over a long period of time. Instead of forcing the user to keep your program running for as long as they want to keep pertinent data, they can store those values into a file that can be read in later. File output is also used when storing user generated content, such as when a user paints a picture in a painting program and then saves it so he can share it with his friends.

In Java 6, file input is handled by `FileReaders`, while file output is handled by `FileWriters`. With these objects you can read and write characters and Strings to files. The resulting file can then be read in a text editor such as notepad. There are also some classes that can read and write raw bytes to files. These classes are `FileInputStream` and `FileOutputStream`. These classes can be used to write more complex data such as image data. All of the file readers and writers can be found in the `java.io` package.

Each file reader or writer can open one file at a time, and each file can only be in "read mode" or "write mode" at any given time. Note that there is no "read/write mode," if you want to read from a file and then write to it, you need to first close the FileReader before you open the file with a FileWriter. If you try to read from a file that doesn't exist, then Java will throw a java.io.FileNotFoundException and if the exception isn't caught, it will crash the program. Another oddity when opening files is how java handles backslashes. Most file paths in windows are written as C:\foo\bar\file but in java the backslash character is an escape character. So when entering backslashes in file names you need to enter it as `C:\\foo\\bar\\file` in order for it to work properly.

Example:

```
import java.io.*;
public static void main(String[] args) throws Exception {

	FileReader fin = new FileReader("C:\\inputFile.txt");
	FileWriter fout = new FileWriter("C:\\outputFile.txt");

	char c = ' ';

	while(fin.ready()) {
		c = (char) fin.read();
		fout.write(c);
	}
	fin.close();
	fout.close();
}
```