1 Reading From a File

Reading from the console is one thing, but sometimes what you have to do is read data from a file. One example of how to do this is shown in Main2, shown below and on the web.

```java
try {
    dataFile = new Scanner(new File("theDataFile.dat"));
} catch (FileNotFoundException ex) {
    System.out.println("ERROR opening inFile " + "theDataFile.dat");
    System.out.println(ex.getMessage());
    System.out.println("in" + System.getProperty("user.dir"));
    System.exit(1);
}
```

There's a lot of boilerplate language stuff in this sample program. The try-catch is the Java construct that allows you to try to execute a line of code, but if a runtime error occurs, then you can catch the error and process it. In this particular example, you try to open a data file of a given filename, but if that file doesn’t exist, you print an error message and exit.

The third version of how to read data, and the second version of how to read from a file, uses static class variables and illustrates how you can clean up your programs by pushing off the error checking into a utilities java file.

The Utilities.java program has two static methods. One of these is the ScannerOpen code from Main2. This is now a public static method, and what public static means is that you can invoke this method as a class method with the Utilities.ScannerOpen(); command. (The class method part is what allows you to use Utilities as the first part of the invocation of that command.)

Note carefully in both Main2 and in Main3 and Utilities what imported files have been requested. These are the java files that include the exception handling code, etc. You will need these.

Note also the difference between the files as produced by linux and by a Windows machine. It is a characteristic of Windows that text files and data files that a given “line” of text ends in a carriage return (ASCII 13 decimal) and a line feed (ASCII 10 decimal). Linux, however, uses only the line feed.
This difference is maintained by Java programs run under Eclipse (actually, I have not tested with programs run from the command line in either environment, so I don’t know if the words “under Eclipse” can be removed from the statement). You will need to be aware of the difference between the two kinds of files.
For example, the output of the Main3 program in the InputOutput2 example is, under Linux, the bytes:

```
0000000  file  sp  version  sp  str
0000020  ing  sp  is  sp  0 9 / 0 1 / 2 0 0
0000040  nl  file  sp  version  sp  s
0000060 tring  sp  is  sp  5 0 0 . 0 0 nl
0000100  file  sp  version  sp  str
0000120  ing  sp  is  sp  balance  enl  f
0000140  ile  sp  version  sp  str
0000160  ng  sp  is  sp  0 9 / 1 0 / 2 0 0 7
0000200  nl  file  sp  version  sp  s
0000220 tring  sp  is  sp  - 2 5 . 5 7 nl f
0000240  ile  sp  version  sp  str
0000260  ng  sp  is  sp  gro c eries  s nl
0000300  file  sp  version  sp  str
0000320  ing  sp  is  sp  0 9 / 1 2 / 2 0 0
0000340  nl  file  sp  version  sp  s
0000360 tring  sp  is  sp  - 1 1 8 . 1 9
0000400  nl  file  sp  version  sp  s
0000420 tring  sp  electric
0000440  ity  nl  file  sp  version
0000460  string  sp  is  sp  0 9 / 1 3
0000500 / 2 0 0 7 nl  file  sp  versi
0000520  on  sp  string  sp  is  sp  1 0 0
0000540 . 0 0 nl  file  sp  version
0000560  string  sp  is  sp  birth
0000600  day  nl  file  sp  version
0000620  string  sp  is  sp  0 9 / 1 4
0000640 / 2 0 0 7 nl  file  sp  versi
0000660  on  sp  string  sp  is  sp  - 7 3
0000700 . 0 2 nl  file  sp  version
0000720  string  sp  is  sp  pre se
0000740  nt  s nl
```
but under Windows you get the extra carriage returns added:

```plaintext
0000000 file sp version sp string sp
0000020 ing sp is sp 0 9 / 0 1 / 2 0
0000040 cr nl file sp version sp
0000060 string sp is sp 5 0 0 . 0 0
0000100 cr nl file sp version sp
0000020 string sp is sp balance
0000140 cr nl file sp version sp
0000160 string sp is sp 0 9 / 1 0 / 2
0000200 0 0 7 cr nl file sp version sp
0000220 string sp is sp - 2 5 .
0000240 5 7 cr nl file sp version sp
0000260 string sp is sp groce
0000300 r ies cr nl file sp version sp
0000320 on sp string sp is sp 0 9 /
0000340 1 2 / 2 0 0 7 cr nl file sp ve
0000360 rsion sp string sp is sp
0000400 - 1 1 8 . 1 9 cr nl file sp ve
0000420 rsion sp string sp is sp
0000440 elctricity cr nl file
0000460 sp version sp string sp
0000500 string sp is sp 0 9 / 1 3 / 2 0
0000520 file sp version sp str
0000540 ing sp is sp 1 0
0000560 file sp version sp string sp
0000600 ng sp is sp birth day cr nl
0000620 file sp version sp str
0000640 ing sp is sp 0 9 / 1 4 / 2 0
0000660 cr nl file sp version sp
0000700 string sp is sp - 7 3 .
0000720 cr nl file sp version sp
0000740 string sp present
0000760 s cr nl
```