

# 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.

```
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 f i l e s p v e r s i o n s p s t r
0000020 i n g s p i s s p 0 9 / 0 1 / 2 0 0
0000040 7 n l f i l e s p v e r s i o n s p s
0000060 t r i n g s p i s s p 5 0 0 . 0 0 n l
0000100 f i l e s p v e r s i o n s p s t r
0000120 i n g s p i s s p b a l a n c e n l f
0000140 i l e s p v e r s i o n s p s t r i
0000160 n g s p i s s p 0 9 / 1 0 / 2 0 0 7
0000200 n l f i l e s p v e r s i o n s p s t
0000220 r i n g s p i s s p - 2 5 . 5 7 n l f
0000240 i l e s p v e r s i o n s p s t r i
0000260 n g s p i s s p g r o c e r i e s n l
0000300 f i l e s p v e r s i o n s p s t r
0000320 i n g s p i s s p 0 9 / 1 2 / 2 0 0
0000340 7 n l f i l e s p v e r s i o n s p s
0000360 t r i n g s p i s s p - 1 1 8 . 1 9
0000400 n l f i l e s p v e r s i o n s p s t
0000420 r i n g s p i s s p e l e c t r i c
0000440 i t y n l f i l e s p v e r s i o n
0000460 s p s t r i n g s p i s s p 0 9 / 1 3
0000500 / 2 0 0 7 n l f i l e s p v e r s i
0000520 o n s p s t r i n g s p i s s p 1 0 0
0000540 . 0 0 n l f i l e s p v e r s i o n
0000560 s p s t r i n g s p i s s p b i r t h
0000600 d a y n l f i l e s p v e r s i o n
0000620 s p s t r i n g s p i s s p 0 9 / 1 4
0000640 / 2 0 0 7 n l f i l e s p v e r s i
0000660 o n s p s t r i n g s p i s s p - 7 3
0000700 . 0 2 n l f i l e s p v e r s i o n
0000720 s p s t r i n g s p i s s p p r e s e
0000740 n t s n l

```

but under Windows you get the extra carriage returns added:

```
0000000 f i l e sp v e r s i o n sp s t r
0000020 i n g sp i s sp 0 9 / 0 1 / 2 0 0
0000040 7 cr nl f i l e sp v e r s i o n sp
0000060 s t r i n g sp i s sp 5 0 0 . 0 0
0000100 cr nl f i l e sp v e r s i o n sp s
0000120 t r i n g sp i s sp b a l a n c e
0000140 cr nl f i l e sp v e r s i o n sp s
0000160 t r i n g sp i s sp 0 9 / 1 0 / 2
0000200 0 0 7 cr nl f i l e sp v e r s i o
0000220 n sp s t r i n g sp i s sp - 2 5 .
0000240 5 7 cr nl f i l e sp v e r s i o n
0000260 sp s t r i n g sp i s sp g r o c e
0000300 r i e s cr nl f i l e sp v e r s i
0000320 o n sp s t r i n g sp i s sp 0 9 /
0000340 1 2 / 2 0 0 7 cr nl f i l e sp v e
0000360 r s i o n sp s t r i n g sp i s sp
0000400 - 1 1 8 . 1 9 cr nl f i l e sp v e
0000420 r s i o n sp s t r i n g sp i s sp
0000440 e l e c t r i c i t y cr nl f i l
0000460 e sp v e r s i o n sp s t r i n g
0000500 sp i s sp 0 9 / 1 3 / 2 0 0 7 cr nl
0000520 f i l e sp v e r s i o n sp s t r
0000540 i n g sp i s sp 1 0 0 . 0 0 cr nl f
0000560 i l e sp v e r s i o n sp s t r i
0000600 n g sp i s sp b i r t h d a y cr nl
0000620 f i l e sp v e r s i o n sp s t r
0000640 i n g sp i s sp 0 9 / 1 4 / 2 0 0
0000660 7 cr nl f i l e sp v e r s i o n sp
0000700 s t r i n g sp i s sp - 7 3 . 0 2
0000720 cr nl f i l e sp v e r s i o n sp s
0000740 t r i n g sp i s sp p r e s e n t
0000760 s cr nl
```