File Input/Output

Most data is stored in files, not input by the user every time. In this activity, you'll learn the basics of reading and writing plain text files.

Manager:		Recorder:	
Presenter:		Reflector:	

Content Learning Objectives

After completing this activity, students should be able to:

- Parse user input and string objects using a `Scanner`.
- Read a text file line by line, and extract data from it.
- Create a new text file, and output several lines to it.

Process Skill Goals

During the activity, students should make progress toward:

- Reading Java API documentation to explore a class. (Information Processing)

Facilitation Notes

Model 1 is an opportunity to introduce `System.in` and `System.out` as files. In that way, students have been doing file I/O since Hello World.

On #3, if the Ctrl+D key doesn't work, have students click the editor window and then click back to the Console window. That should reset the Console input to recognize Ctrl+D again. Students might confuse their own input as part of the program's output. Make sure they don't have two copies of each line in their answer.

At the end of Model 2, compare answers for #8 and #12. Discuss how to handle `FileNotFound` exceptions (e.g., print a stack trace, exit with an error code, or ask the user to try again). Remind students of the difference between `next()` and `nextLine()`.

During Model 3, have students open their *results.tsv* file both in a text editor and a spreadsheet application. Discuss the file format and how it's displayed in each application.

Key questions: #5, #12, #14 Sample solution: *IMDb.java*

Source files: *ScannerDemo.java*, *title2020.tsv*

Model 1 Review of Scanner

The `java.util.Scanner` class is useful for reading and parsing text from various sources:

```
// Example 1
Scanner in = new Scanner(System.in);
while (in.hasNextLine()) {
    String line = in.nextLine();
    System.out.println(line);
}

// Example 2
String text = "1 fish 2 fish red fish blue fish";
Scanner sc = new Scanner(text);
System.out.println(sc.nextInt());
System.out.println(sc.next());
System.out.println(sc.nextInt());
System.out.println(sc.next());
```

Questions (10 min) Start time:

1. For each example above, describe what the `Scanner` is scanning.

a) Example 1: `new Scanner(System.in)` keyboard input

b) Example 2: `new Scanner(text)` the "fish" string

2. Based on the documentation for `Scanner`, explain the following:

a) `in.hasNextLine()`

Returns true if there is another line in the input of this scanner.

b) `in.nextLine()`

Advances this scanner past the current line and returns the input that was skipped.

c) `s.nextInt()`

Scans the next token of the input as an int.

d) `s.next()`

Finds and returns the next complete token from this scanner.

3. Open *ScannerDemo.java* in Eclipse, and run the program. Enter three lines of input, and notice the output. Then press Ctrl+D, which is the keyboard shortcut for "end of file" (EOF).

a) In the Console, what color was the user's input? blue/green

b) In the Console, what color was the program's output? black

c) What was the complete output of the program? (Note: Do not include the input lines.)

```
==== Example 1 ====
Line 1
Line 2
Line 3
==== Example 2 ====
1
fish
2
fish
```

4. What effect did pressing Ctrl+D have on the program? Explain how you think EOF works.

The first example repeated while the input had a next line. EOF caused the program to exit the while loop and move on to Example 2. Once the keyboard input "file" ended, `hasNextLine` returned false.

5. Rewrite the code for Example 2 to output each *word* of the string using a `while` loop. Run your code to make sure it works.

```
Scanner sc = new Scanner(text);
while (sc.hasNext()) {
    System.out.println(sc.next());
}
```

Model 2 Reading from a File

The Internet Movie Database (IMDb) maintains information about movies, television shows, video games, and more—including their cast, production crew, trivia, ratings, etc.

Download the *title2020.tsv* file into your Eclipse project. It's a subset of all titles from the year 2020 and is based on the data available at https://www.imdb.com/interfaces/.

Create a new program named *IMDb.java*, and add the following lines to the `main` method:

```
File file = new File("title2020.tsv");
Scanner in = new Scanner(file);
for (int i = 0; i < 3; i++) {
    System.out.println(in.nextLine());
}
in.close();
```

Note: You will need to import `java.io.File` and `java.util.Scanner`.

Questions (20 min) Start time:

6. Explain the compiler error when attempting to construct the `Scanner`.

Unhandled exception type FileNotFoundException.
(The constructor throws FileNotFoundException.)

7. Explain two ways you can modify the code to handle this error. (*Hint:* Eclipse offers them as "quick fixes".) Which way is better?

You can (1) add a throws declaration, or (2) surround with try/catch. The first option is simpler because it changes the least amount of code. But it causes the entire method to ignore file not found exceptions. The latter option is better, because it isolates the problem.

8. Modify the program so that it compiles: 1) surround the "`new Scanner`" line with try/catch; 2) remove the auto-generated `TODO` comment; and 3) initialize the `in` variable to `null` (before the `try` block). Copy and paste the beginning on your `main` method (from the "`File file`" line to the end of the `catch` block) in the box below.

```
File file = new File("title2020.tsv");
Scanner in = null;
try {
    in = new Scanner(file);
} catch (FileNotFoundException e) {
    e.printStackTrace();
}
```

9. Run the program. What is the output of the `for` loop?

```
tid type    title   original    isadult year    ended   runtime genres
60366   short   A Embalagem de Vidro    A Embalagem de Vidro    0   2020        11  Docume
62336   movie   El tango del viudo y su espejo deformante   El tango del viudo y su espejo
```

10. TSV stands for "tab-separated values". Explain the format of the *title2020.tsv* file:

a) What does the first line represent? the attribute names

b) What do the remaining lines represent? movies / TV shows

c) How are "column breaks" represented? by tab characters

11. By default, `Scanner.next()` separates input by whitespace. Replace the `for` loop in your `main` method with the following code. What is the resulting output?

```
in.useDelimiter("\t");
in.nextLine();

for (int i = 0; i < 3; i++) {
    System.out.println(in.next());
}
```

```
60366
short
A Embalagem de Vidro
```

12. Remove the `for` loop from your program. Write code that outputs the `tid`, `type`, and `title` of the first 5 titles that start with `"A"`. (*Hint:* Use a `while` loop, call `in.next()` to get each of the three values, and call `in.nextLine()` to advance to the next line.) Paste your code below, starting from the `while` loop.

```
int count = 0;
while (count < 5) {
    String tid = in.next();
    String type = in.next();
    String title = in.next();
    if (title.startsWith("A")) {
        System.out.println(tid + "\t" + type + "\t" + title);
        count++;
    }
    in.nextLine();
}
```

Model 3 Writing to a File

The `java.io.PrintWriter` class is useful for writing text files:

```
File file = new File("results.tsv");
PrintWriter out = new PrintWriter(file);
// output text to the file...
out.close();
```

Questions (15 min) Start time:

13. Examine the documentation for `PrintWriter`. What methods can be used to output a string to the file?

append, format, print, printf, println, write

14. Modify your code from Question #12 to output to the *results.tsv* file instead of to the screen. Summarize your changes below:

1. Add the lines from Model 3 to the beginning/end of `main`.
2. Surround the PrintWriter constructor in a try/catch block.
3. Replace each `System.out.println` with just `out.println`.

15. In general, is it easier to write code that reads a file or writes a file? Explain your reasoning.

Writing a file is much easier, because you don't have to parse the file contents and deal with potentially incorrect formatting.

16. Make sure the end of your `main` method closes both files. Why is it important to close files when you are finished with them?

Closing a file releases any system resources associated with it. And when writing files, the output may not be on disk until it's closed.

17. (Optional) What is the difference between the `print` methods and the `write` methods in the `PrintWriter` class?

The `print` methods should be used most of the time for outputting text. The `write` methods are lower level and used for printing characters without encoding them based on the current operating environment.