Model 1 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:

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

append, format, print, printf, println, write

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

1. Add the lines from Model 1 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`.

3. 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.

4. 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.

5. (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.