CS 116X/106 **Lab 01** **Winter 2015**

Writing and Reading Files

Useful Resources:

```
split()
splitTarget()
global
loadStrings()
createReader()
createWriter()
readline()
noLoop()
```

Instructions:

Be sure to save any attached files (that are being read or written to) in the same folder as your code. All parts of lab questions that have **R** at the beginning are mandatory, and their submission by each student is required. The parts of each question that begin with **O** are open-ended enhancements for further investigation into course material. These should be completed only after the mandatory parts of the lab are completed, and are not explicitly required for submission. If you choose to complete further optional add-ons, you are welcome to explore your own options, and not just the suggestions provided. Submit a zip document to the dropbox folder L1 on LEARN. The document should contain separate folders for the following files:

- **L1_Hamlet**
- **L1_HamletWords**
- **L1_BarChart**

Writing Files Example

The provided Python file DrawBars.pde is a simple example of how to write code to a file you create. In this example, running the program will produce a grid where you can move your mouse to create a bar of the height of your mouse. When you press the mouse, the program writes that current value to your new program file. When you press 'Q' the program ends, and a file named output.txt will be created, or updated if the file already exists. The file contains all the mouse values of y in the order that you clicked while running the program.

Reading Files Example

The provided Python file Average.pde is an example of how a program can read the data in a text file. In this example, when run with a input.txt file with a number on every line, the program will print the average of all the numbers within the input.pde file. Any output.txt file from DrawBars.pde could have its name changed to input.txt and run through this sketch.

---

1. Hamlet

a. **R** Download the provided text file soliloquy.txt. Create a sketch that reads soliloquy.txt and prints each line of the file using `println()`. **Hint:** You code should be able to run regardless of what is in the file, as long as you use `loadStrings()`.

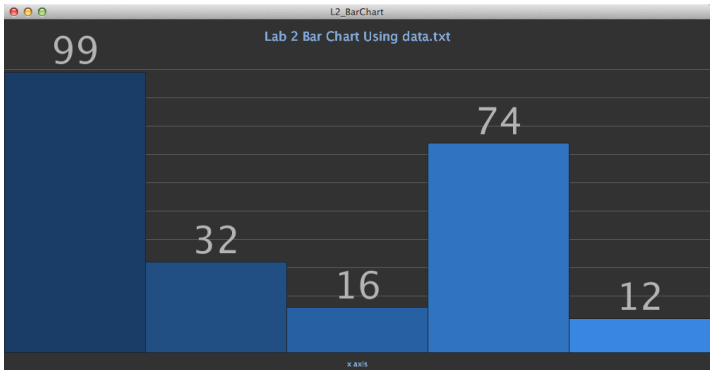
2. HamletWords

a. **R** Copy your code from L2_Hamlet into a new Processing Sketch. Modify your code to print each *word* of the file soliloquy.txt to a new line. For this question, we will define a word as a sequence of characters (including punctuation) bounded by spaces or new lines.

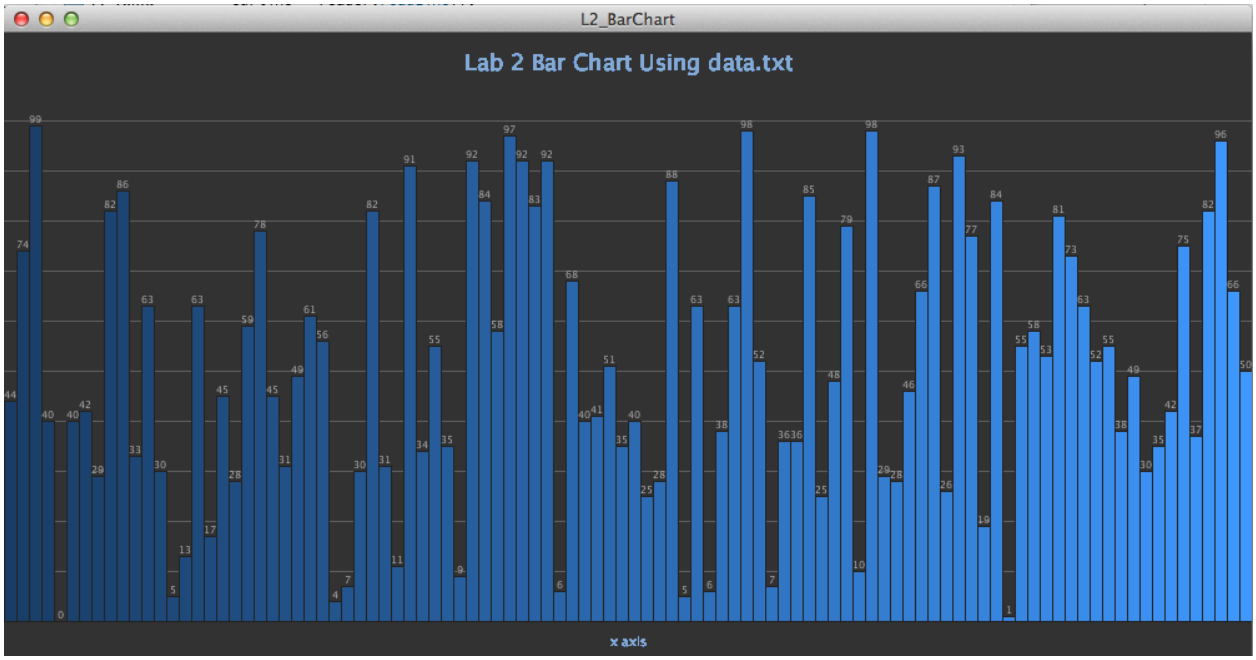
b. **O** Speed-reading is a new technological phenomenon you can find at http://www.spritzinc.com/. Update your code to act like a basic speed-reading app. Show each word in soliloquy.txt one at a time centered on the canvas.

2. BarChart

a. **R** Using the provided data.txt file, write a function that creates a simple bar chart by reading the file, and outputs each line of data (a number between 0 and 100 inclusively) as the entry in a bar chart. You may assume that data.txt will have no more than 100 lines. For example, the text file containing the numbers (99, 32, 16, 74, 12) on separate lines would output the following bar chart.



b. **O** Add on each bar of the graph the number that corresponds with the height of the bar. Also add a title to the graph.


L2_BarChart
Lab 2 Bar Chart Using data.txt
x axis