# Lab 2: Textual Analysis in Ruby

Author: Nigel Nelson
Course: CS2040
Date: 3/20/21

What did you learn from doing this lab?

- My biggest take away from this lab is how to use blocks in Ruby. I learned that it is preferable to use blocks because it can lend the ability for operations to be vectorized, instead of requiring for every element to be iterated over like in a traditional loop. This style reminded me a great deal of functional programming in Python, so I really enjoyed this type of coding. In addition, I learned how to create and use classes in Ruby. Previously we had only used unorganized code in Ruby, but by using classes and methods the responsibilities of a program can be more efficiently divided so that code is much more cohesive and also more reusable.

What did you find challenging about the lab?

- My biggest challenge for this lab was using blocks instead of traditional loops. Even though my past quarter required a great deal of functional programming in Python, many of the text processing problems that I was tasked with solving in this lab came much easier to me when I thought about them in terms of traditional loops. As such, it took me quite some time to rework solutions so that instead of loops I could use ".each" and ".map" to iterate over elements. Another portion of this lab that was difficult for me was just the absence of an IDE. This is because in object-oriented design with an IDE, it will let you know if you mislabeled a variable, or a class reference, so that it is easy to correct. However, when coding in Notepad I quickly realized I had to be very meticulous and careful in my coding because I would not encounter noticeable errors until I attempted to run my program.

What would you recommend changing if this lab is reused in future years?

- There is not much that I would change from this lab. I believe the requirements are very clear, and there are many input and output examples so that students know what is expected of their final program. Perhaps the one thing that may be helpful is to encourage students to find a Ruby IDE of their choice. This is because I spent a solid portion of time debugging simple syntax errors that were a result of accidently hitting a key, and I believe that if an IDE caught this, I could focus more on the nuances of coding in Ruby and less on the nuances of coding in Notepad.