

Reflection Report – A2

For A2, I chose to do the first assignment: a user enters a filename, and then the program will scramble words from that file. To build this program, I used the following steps.

1. Build skeleton of program. This includes creating stubs for all required functions and developing an algorithm for parsing the file.
2. Build a helper function to check if the file exists so that it can be opened for reading.
3. Build the function to get user input for filename, and check if it exists using above function.
4. Once the file was verified to exist, begin parsing the file. The algorithm to process it goes as follows:
 - a. Read the file line by line, starting at the first line.
 - b. Keep track of the beginning index of the word.
 - c. Read through the contents of the line, keeping track of how long the current word is.
 - d. Once a certain character is met letting us know this is the end of the word, split the line from the start index and length of the word to extract it from the line.
 - e. Update the start and end index to begin reading the next word in the line.
5. Build isWord (only alpha characters allowed, so words with a ' and a number in it are not true words) and RNG function (uses a randomizer package generate a random location for a letter to be placed)
6. Determine if a substring should be processed for randomizing in an if statement using isWord.
7. Build scramble word function. Scrambles it by ignoring first and last chars, then keeping a running list of all positions previously used and using the RNG function. Build the new word and return it.
8. Print out the results of the scrambling. If the word was not processed, it is still printed.

Overall, I found that Ada was well suited for developing this program, although there were some limitations of Ada that I did not enjoy working with. To begin, I really liked working with exceptions in Ada. Exceptions made checking certain things such as if a file was properly opened simple and easy to code. If I did something like that in C, it would have taken more time, making Ada a good choice. I also found Ada well suited for working with file IO such as opening, reading the contents of the file and closing a file. The process was streamlined and easy to learn, making reading the contents of the file trivial so that I could focus on the processing algorithm.

Although working with these structures made me enjoy coding in Ada, I ran into many bumps along the way. A major downside I found was working with strings. I did not enjoy the process of using unbounded strings, as I found them a bit limited in their functionality and finicky. I had to convert between unbounded strings and normal strings quite frequently, and it became a pain to manage them separately. Another thing I found a bit difficult to deal with in Ada was the structure of the application. I initially had trouble organizing the flow of my program, as sometimes I found I wanted to declare variables in the middle of a function, but Ada prevented me from doing that, so I had to think of workarounds instead.