Reflection

For this assignment I chose to write the wordscram.adb script. In doing this a multitude of design decisions were made from how functions would work to where printing should be. My first design choice was to make all of my printing except for the number of processed lines happen in processText. This was because I figured that the place where the text was read in would be best to print the scrambled text. Another design decision I made was in the randomInt function where the lower bound was statically set to the value 2. This was done as the first letter was always skipped. With this being said the upper bound was dynamic to reflect the length of the word that was scrambled. A third design decision that was made was with the number of processed words. Since the wording was vague I decided to consider any word that went through scrambled word to be processed. This means that any word less than 4 letters would not be considered processed and wouldn't increment the counter. My thought process was that since isWord and my check for length >3 was in the same if then they weren't really processed as the words were never changed. Overall, I believe that Ada was well suited to solve this problem as it easily handles strings and manipulation of them. The ability to convert strings to chars and edit them combine with the use of functions makes the program efficient and effective. In addition to this the ability to create types such as the random and int_arr types I made allowed for more unique data structures to be used. This saved a lot of space and made the program that much easier to write. As for the Limitations with Ada there seemed to only be a few. One of which was that all functions seemed to need to be nested inside the process function to work. This was a little annoying and made the program a bit more tedious to read. In addition, the fact that strings are not entirely easy to return from functions as the return value size must be the exact same as its return value made it a bit annoying.

Design approach:

1. Wrote getFilename()
2. Wrote main to getFilename()
3. Wrote processText()
4. Added error checking to getFilename()
5. Wrote scrambleWord() body
6. Wrote isWord()
    a. Connected isWord to scrambleWord() using processText()
7. Wrote the rest of scrambleWord()
8. Wrote randomInt()
9. Wrote catch for words <3 in processText()
10. refactored