**Morse Code Translator**

FINAL PROJECT

**Part 1**Write a program which will read a file from the disk called "morse\_code\_text\_to\_translate.txt". This file will contain words and numbers. Your program will read the file, convert the input text into Morse code and write the Morse code into a new file called "morse\_code\_translated.txt".

You can find information about Morse code at the following: <http://en.wikipedia.org/wiki/Morse_code>

<http://ling.ucsc.edu/~hank/morseabc.html>

Use a period '.' For the dots and a dash '-' for the dashes. Put a space after each set of dots and dashes that make up each letter so that you can read the file back in for part 2 of the program.

For example – the input text "ruby rules" will be represented by (note the spaces between words):

“.-. ..- -… -.-- .-… .-. ..- .-.. . …”

Each “word” of Morse code characters will be separated by a “wait” which is represented by “.-…” I realize this is non-standard morse-code, but morse-code has no real concept of a "space".

Note: You must take into account text that is lowercase since Morse code does not make that distinction.

**Part 2**

Write a program that reads the output file from part 1 – “morse\_code\_translated.txt”. Convert each Morse code sequence back into a letter and save the converted / translated text to a file called “translated.txt”. These 2 files should be able to be converted back and forth with no errors.

Note that each Morse code “character” (sequence of dots/dashes) will be delimited by a space for easier translation. Do NOT put the spaces into the letters that you save back to the file. You need to take into account the "wait" as mentioned above and convert that into a space when writing it back to the file.

You will have to handle both text and digits when reading/writing the code and translation.

Use the 2 files that I give you for testing your input/output. Your program **MUST** use methods for the various pieces of code.

HINT: Think about your how the data is stored, and how certain data structures allow you to index various elements very easily.

HINT2: I have included a file called "MorseCodeAlphabet.txt" which is a file suitable for reading into your program for use in a data structure as appropriate.

Turn in both programs (or one single program if you have a menu option). Name the file "morsecode\_yourname.zip" or "morsecode\_yourname.rb" if only one file. Upload as an attachment to the assignment in Canvas.

NOTE: You do not have to turn in your text files, unless you have an initialization file to create your translation structure. I will run this on my own files and create the files from your program.