As of April 22, 2021

**User Manual / Product Documentation**

Since this product is designed as a developer tool with the intended user being a developer or someone with a similar background and knowledge, it will be assumed that the user has an understanding of basic software development, and understating of general computer knowledge, and how to navigate and use their own OS and File Systems. It is also assumed that the user will be able to reconfigure various path files for their own systems.

Because it is a developer tool, the most common use of this is to simply only use a few parts of the entire program to meet the needs of whatever the developer does. It is unlikely that a developer will have the need to go through the entire program and use all of its pieces.

The program is designated as a multi-level menu screen and the user is free to use it this way, however, when the user becomes more familiar with the tools, they may find it more efficient and helpful to simply directly use the tools without having to navigate to them through all of the layers of menu.

Since this project has a lot of room to grow after the scope of this class, there are several areas in the menu navigations screens displayed to the user that describe some dev tools or features as “coming soon” or “still in development”.

**Dev Tool 0 – Base Tool Information**

The base tool is a collection of functions that make shortcuts for common programming practices and needs.

It includes the following methods/ features to perform the following operations:

* addDefinition(string term, string definition)
  + Accepts a term and a definition to add to the custom dictionary
* addWebLink(string title, string link)
  + Accepts a title/description of a website or webpage as well as a link to that site to be stored in a file
* appendLineToEOF(string fileName, string text)
  + Accepts a name of an existing file including the extension and the text that is to be appended to the end of the file
* capitalizeWord(string text)
  + Accepts a word and returns it with the first letter capitalized
* createFile(string fileName)
  + Accepts a name of a new file to create along with its extension and then creates the file
* createClassHeaderFile(string fileName)
  + Accepts the name of a new class header with the extension, for example: “MyNewClass.hpp”. Then prompts user to enter the name of the class and the variables a long with their data types. Once the user is finished, it then generates a new HPP file where ever your machine creates new files (XCode of course being in an odd place and not right in the local project).
* getFileLineCount(string fileName)
  + Accepts a name of an existing file including the extension and returns the number of lines in the file
* lookUp(string term)
  + This method queries the Dev Tool Custom Dictionary for the term passed in as a parameter. Returns the definition of the term if it exists or returns a not found message.
* readFile(string fileName)
  + Accepts a name of an existing file including the extension and returns a vector with each item in the vector being a line of text from the file.
* textToSpeech(string text)
  + Accepts a string of text to write to a file. The file is then read allowed by the system commands.

**Finding and Configuring Files and Project Parts**

This project has a large number of external files. Due the variation in where files are stored on different machines, with different IDEs, or different compiler/processing behaviors, a more detailed explanation is required. Inside of the main project folder (where the main source file and other CPP and HPP files are located) there are 4 additional folders:

* Audio Files – this folder contains the mp4/wav files for relaxing music. All though they are directly sored in the same location as the other files of this project, the path currently written in the Dev 5 code is an absolute path based on my personal system and would need to be configured for your own system.
* Arduino Files – Similar to the audio files folder, this folder is stored directly in with this projects files but because it used the Arduino CLI which is on a different location on my machine, you will also need to configure to match your system.
* Python Files – This folder contains 3 files. One is a Python file that is used to web scrape information from a website, format the information and produce two files from the formatted text. This file is not meant to be run in the project, but is there to show how the files were generated 1 time to be used in other parts of the C++ project files.
* Other files – these are a copy of the files that would normally be generated directly in the same directory as other project files on Windows machines. But XCode IDE stores them in a different place outside of the project. These files are required for the functionality of much of this product so they will need to be moved to where ever the files are stored with in your machine/IDE. They directories will also need to be changed with in the source code to reflect your machine/IDE.

Other requirements/ important information:

* This product was designed on and for a mac OS. Using it with other operating systems could result in errors and unpredictable behavior
* Parts also rely on the addition of an Arduino microcontroller as well as Arduino software requirements and the Arduino CLI tool. The software components can be found at: <https://arduino.github.io/arduino-cli/latest/commands/arduino-cli/> and <https://www.arduino.cc/en/software> as well as other pages on the Arduino official website