## Jargon Generator Description

### Overview:

The jargon-gen.exe program creates short lists of phrases. It does this using three separate input files, two containing adjectives and one containing nouns. An example might be: “big black cat”.

The application jargon-gen.exe is a self-contained executable, no install is required. However, there is a requirement for three lists of words to be contained in the working directory. list-1-adj.txt, list-2-adj.txt and list-3-noun.txt are required and included in the zip file.

For developers wanting to improve on the game the source code is included in the \source folder and includes:   
- main.cpp: the main program containing program flow  
- Jargon-Gen.cpp: all program functions (excluding main)  
- Jargon-Gen.h: function definitions and program constants for the application

### Function Descriptions:

### main

The main function controls the program flow for the application. The main function calls initialise to seed the random number generator, it then calls a screen display function called main\_menu to describe to a user how to work with the program. The main\_menu function is called whenever an operation is complete so that users are always presented with choices for the application. The main function uses the first user input to decide which function to call next with the decision being made inside a series of if then decisions.

### initialise

Function to seed the random number generator. No input or return values. The rand function will be used to choose a word from the lists. In this function the srand() function is used to seed the rand() function used later. It is defined in stdlib.h, and this library has been included in Jargon-Gen.cpp.

**main\_menu**Function for screen display only. Used to present the user with choices for the application and called after every successful operation.

**bad\_choice**Function for screen display only, called whenever bad user input is detected (input that is not expected), directs user back to main menu after alerting them of error condition.

**quit\_application**Screen display to advise users they are closing the application. Application terminates here.

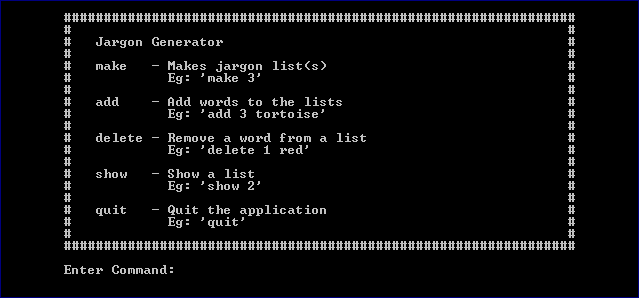
**Add\_entry**  
Function to add a new word to the list of words. The function uses ofstream to write the new word to the output file. It accepts an input telling it which list is to be modified.

**Delete\_entry**  
Function to delete an entry from one of the lists. It uses ifstream and ofstream to read the file into an array, then return the data to the file minus the word the user has requested deletion of. It accepts input parameter saying which list the word should be in.

**Show\_list**  
Function to display to the user what is in the list that they are interested in. The function accepts a list number as a parameter. This function can be used to check if the add\_entry or delete\_entry function has been successful.

**list\_make**  
Function to generate random jargon lists. The user types the number of lists they require and this is passed as a parameter to the function.Each file is opened, read into an array, then the random number function picks a word at random from each array and constructs a jargon phrase.

## Sample Program Output

Program Launch:

Make command:

