**Joshua Wilkes  
Programming Languages  
Project 2  
User’s Manual  
  
Setup and Compilation**1. Download and unzip the submission from eLearning on a Linux box in the multi-platform lab.  
2. The submission includes:

* declaredIdentifiersList.c
* declaredIdentifiersList.h
* errorList.c
* errorList.h
* FunctionalDecomposition.docx
* lexicalAnalyzer.c
* lexicalAnalyzer.h
* main.c
* makefile
* parser.c
* parser.h
* prog1legal.txt
* prog1Illegal.txt
* prog2legal.txt
* prog2Illegal.txt
* prog3legal.txt
* prog3Illegal.txt
* prog4legal.txt
* prog4Illegal.txt
* registerBuffer.c
* registerBuffer.h
* registerStack.c
* registerStack.h
* simpleProgramParser
* tokenList.c
* tokenList.h
* UsersManual.docx

3. Environment: This program has been tested in the multi-platform lab and will run there.

4. Compiling. This step not necessary as the script to run the program will compile it. If desired, however, this program includes a Makefile. At the command line in Linux, type “make”. The program produces an executable entitled “run”. The command “./run <filename>” will execute the program for the given filename.

**Running the program.** Be sure that no files have been moved from the given folder. The script is called “simpleProgramParser”. Execute the command “./simpleProgramParser” to compile and run the program for all eight text files. No command line arguments are required for the script, nor are they checked.

User input: no user interaction with the program is required.

**Output:** Initial output is directed to the console to show that the program ran successfully. Register output is directed to files that get generated by the program. The files are also in the script to be displayed on the console using the “cat” command.