Programming Languages

Dr. Gurka, Spring 2019 Project #1 – MAL Syntax Checking

January 28

Extra Credit Ideas

You may do any one or more. Remember to list and discuss extra credit in your cover memo, and you may work with a classmate (on the extra credit only). For any you do, give explicit specifications as to what you are detecting, if more detail is needed that what is given here. Make one or two new MAL programs for your extra credit; include that and its report file(s) in your final packet. Note: the code additions should take only about an hour or two (each) to code and test. Extra credit is due with the final on-time project; no extra credit on late projects.

1. Check that the END instruction appears exactly once in the program, and is the last instruction in the program.
2. Process all errors in one line of MAL code, instead of stopping at the first error found. Provide multiple error messages in the report file. (Not always possible – why?)
3. Generate a list of identifiers, such as would be used to create a symbol table; give a list of these at the end of your report file. Do not show duplicate identifiers in the list. Registers do not count as identifiers.
4. Collect and report some fairly primitive “style” stats on the code itself, regarding white space and comments. Use one of your two enhanced MAL programs, but show a new report file. Pick one or both of …
   1. Count the number of code lines (with or without on-line comments) and blank lines and comment-only lines as you process the MAL code. At the end of the report file, report each count followed by its percentage of the total number of lines.
   2. At a character level, count percentage of comments vs. code (do whatever you want with white space characters, just explain it) and report it in the report file.

-----