**Introduction**

The Goal of the project is to develop a simple compiler -- more specifically understand and implement key steps in developing a compiler. In this project you will implement a compiler for the [LITTLE programming language](https://drive.google.com/open?id=0B2Ja5B1pmMV3bWhvaDgzb2xMaEk). It is a toy language based on, but different to MICRO language used in the text book. It is based on the MICRO language used in the “dragon book” [1].

The project will be composed of four main steps (scanner, parser, symbol table and semantic analysis). A set of inputs and the corresponding outputs will be provided for each step. You are free to use any language/tool/platform in order to develop the compiler. There will be deadlines for each step, and you are expected to submit an executable that takes the name of an input file from the command line and writes the output to the screen. The unix *diff* will be used to compare the output from your code with the original outputs for grading.

Unfortunately there will not be any technical support. For the project component you are expected to work together and submit just one final product.

**Grading**

All points from 4 steps will add up to 80 points and are worth 80% of the final grade. All submissions will be tested against a set of hidden/published test cases.

**Running your compiler**

You are expected to submit original source files/libraries and a means for building it (for e.g. a makefile) -- please don’t submit just the executables. Also, if you are using a tool to generate source files do not submit the tool-generated source files, instead you should provide the original source files and ensure that the Makefile calls the tool as part of its build process.

The grader may use the following commands to build and run your compiler (for e.g. in C/C++).

make clean; make compiler

./Micro test.micro > <output file>

Note that the input is being read in from a file passed as a command line argument, not redirected from STDIN, while the output is being redirected from STDOUT.

Bottom Line is please submit all necessary files for building your compiler and a short readme file with instructions.

All files relates to each step will be kept in its own subfolder (e.g. “Step 1”)

**References**

1. Aho, A. V., Sethi, R., & Ullman, J. D. (1986). *Compilers, Principles, Techniques*. Addison wesley.
2. Andrew, W. A., & Jens, P. (2002). *Modern compiler implementation in Java*.
3. Fischer, C. N., Cytron, R. K., & LeBlanc, R. J. (2009). *Crafting a compiler*. Addison-Wesley Publishing Company.