**Course overview**

This course will be an independent study (CSCI 492) that is an equivalent of CSCI 468 (**Compilers, 4 Credits)**. The objective will be for you to gain at least the amount of knowledge/experience you would have gained had you taken the regular CSCI 468.

Course grade is computed from **a** **project (80%) and a final report (20%)**. There will not be any exam(s). The Goal of the project is to develop a simple compiler -- more specifically understand and implement key steps in developing a compiler. You will be developing a simple compiler for the toy programming language called “LITTLE” language. 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.

**The report** should have the structure of a typical scientific paper composed of an introduction, background, Methodology (data/method/tools etc), Results and discussion, Future work, References. It should consist of at least 20 pages. For the report component you are expected to work together and submit just one final report.

Although the project is well-structured I’m flexible to adjust details as you move forward. The most important aspect I will be looking for is your maximum effort.

**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.