**CSC306 - Software Development in C++**

**Final Project Part 3: Compiler – Due Friday, May 31st by 2:40pm**

This is a group project. You are required to work with two other members in class. Groups will be assigned in class.

The goal of this part of the compiler project is to build an instruction table and then execute the instructions in that table. Your group's program must use the code attached. You have already implemented the scanner and parser. This part of the assignment assumes those other parts are working, so you can start with a valid token/lexeme file, similar to how part 2 started. We will review the specifics of the project in class. Make sure you take notes

A few requirements:

* You must submit a complete working program as a single .cpp file. If your code contains syntax errors (these do not include NetBeans errors), you will receive an automatic 0. So, make sure you comment out any code that is triggering syntax errors.
* Your group must turn in a hard copy at the start of the last class (Friday of week 10). Each member must turn in a group evaluation form (individually). If a member does not submit a form, they will lose 20% points (other members will not be affected).
* Each member must upload the final version to Blackboard AND a copy of their individual test plan for their methods. The test plan must include three different test cases. Each test case must include a source code sample testing a particular condition for their methods, the token/lexeme list that matches the source code, and a symboltable for the code.
* Your program will read in a token/lexeme file and a separate symbol table file. You may assume the token/lexeme file represents a syntactically valid piece of code based on the grammar from part 2 of this multi-part assignment. In other words, assume it passed the lexical analysis and syntax analysis steps.
* Your program must handle all possible source code examples that follow the grammar provided in the previous assignment (part 2 of this compiler project).
* All methods written by your group must include pre/post conditions and the name of the group member responsible for writing the method/function. Review your final completed version to ensure duplication of code is minimized.
* Code is assigned as follows:

Member 1: AssignStmt, IfStmt, ConstExpr, Compile

Member 2: InputStmt, ExprOutStmt, IdExpr, Run

Member 3: StrOutStmt, WhileStmt, InFixExpr, dump

All Members are responsible for writing the methods in the Compiler constructor and writing main.