**CPSC 323 Documentation**

**Assignment 1: Lexical Analyzer**

**Randy Le**

**Alex Ma**

# Problem Statement

<write the problem statement here. You can mostly get it from the assignment itself>

The goal for this assignment was to create a lexer that returns a token when it is needed. The lexer returns a record that has tokens in one field and the values of that token (instance of a token) in the other. One of the main requirements of this assignment is that we had to construct a FSM. Our main reads in a file and outputs the tokens and values. The output file has Tokens in the first column and lexeme in the second column.

# How to use your program

# *<write detailed steps how to execute your program>*

The program can either be ran by the executable or by directly compiling in visual studio. The files provided are compatible on Windows 10 as tested in the lab. Since this is the first part of the compiler, only the lexer is functional.

# Design of your program

< write major components of your program. Also, data structures you are utilizing, particular algorithms you have chosen etc. >

We created a class called LexerAnalyzer that helps us parse the input and determines if the given text is a keyword, separator, operator or identifier. We read in the input from the file and then output the data correctly with the tokens and lexeme. How it works Is we call the lexer for a token then print the token and lexeme. This is done though a while loop.

# Any Limitation

<All features are running according to the assignment, but you limit your program due to resource limitations, such as

*Maximum number of lines in the source code, size of the identifier, integer etc.*

***Say ‘None’ if there is no limitation****>*

# Any shortcomings

*<Anything you could NOT implement although that is required by the Assignment.* ***Say ‘None’ if there is no shortcoming****>*