**CSC 426 - Compilers**

**YASL Definition for Project #2**

**Expression Parser Rules**

This definition uses the same tokens from project #1 (the YASL definition handout includes ALL tokens, but you should only need to recognize the ones from project #1 for this project).

The grammar rules for the subset of YASL you need for this project are as follows. The notation here encloses variables in < > brackets. The terminals are tokens; the assumption is that for keywords and operators, the token name matches the name of the keyword or operator. If the expression is not clear, please ask.

<Program> -> PROGRAM ID SEMICOLON <Block> PERIOD

<Block> -> <ConstDecls> BEGIN <Statements> END

<ConstDecls> -> <ConstDecl> <ConstDecls>

<ConstDecls> -> 

<ConstDecl> -> CONST IDENTIFIER ASSIGN NUM SEMICOLON

<Statements> -> <Statement> <Statements>

<Statements> -> 

<Statement> -> PRINT <Expression> SEMICOLON

<Expression> -> <Expression> PLUS <Term>

<Expression> -> <Expression> MINUS <Term>

<Expression> -> <Term>

<Term> -> <Term> STAR <Factor>

<Term> -> <Term> DIV <Factor>

<Term> -> <Term> MOD <Factor>

<Term> -> <Factor>

<Factor> -> NUMBER

<Factor> -> IDENTIFIER