**CSC 426 - Compilers**

**YASL Definition - Lexical Rules**

YASL has a number of different legal tokens - they are enumerated here:

* **Identifier (ID)**

An identifier is a letter followed by zero or more letters (case is significant) or digits.

* **Number (NUM)**

A number is a non-zero digit followed by zero or more digits, or a zero digit (not followed by any digits).

* **String (STRING)**

A string is any set of zero or more characters other than a double quote “, surrounded by double quotes; if a string literal needs to contain a double quote it should be doubled “”

* **Keyword**

The keywords in YASL are program, const, begin, print, end, div, mod, var, int, bool, proc, if, then, else, while, do, prompt, and, or, not, true, and false.

* **Punctuation**

The punctuation in YASL are semi-colon ;, period ., colon :, left parenthesis (, right parenthesis ), and comma ,.

* **Operator**

The operators in YASL are plus +, minus -, star \*, assign =, equal ==, not equal <>, less than or equal to <=, greater than or equal to >=, less than <, and greater than >.

* **End-of-File (EOF)**

At the end of the input file, an end-of-file token should be generated.

Tokens in YASL may be separated by zero or more whitespace characters (space, tab, carriage return, or newline) or comments.

There are two forms of comments:

* Starts with a left curly brace { and runs up until the first following right curly brace }. Any character other than } (including newlines and carriage returns) may be in the body of the comment; this means they can span multiple lines but cannot be nested.
* Starts with two forward slashes //, and runs up until the end of the current line; any characters other than a newline may be in the body of this comment.