REDUCED TRIANGLE DEFINITION FOR SCANNER & PARSER

NON-TERMINALS

Program	::=	Command
Command	::=	Single-Command (;single-Command)*
Single-Command	::=	V-name (:=Expression|(Expression))
	|	**if** Expression **then** Single-Command
		else Single-Command
	|	**while** Expression **do** Single-Command
	|	**let** Declaration **in** Single-Command
	|	**begin** Command **end**
Expression	::=	Secondary-Expression
	|	**let** Declaration **in** Expression
	|	**if** Expression **then** Expression **else** Expression
Secondary-Expression	::=	Primary-Expression
	|	Secondary-Expression Operator Primary-Expression
Primary-Expression	::=	Integer-Literal
	|	Character-Literal
	|	V-name
	|	Identifier **(** Actual-Parameter-Sequence **)**
	|	Operator Primary-Expression
	|	**(** Expression **)**
V-name	:==	Identifier
Declaration	:==	Single-Declaration (; Single-Declaration)*
Single-Declaration	::=	**const** Identifier **~** Expression
	|	**var** Identifier **:** Type-Denoter
Actual-Parameter-Sequence	:==	(Actual-Parameter (,Actual-Parameter)*)
Actual-Parameter	::=	Expression
	|	var V-name

Type-denoter	::=	Identifier

TERMINALS (MICROSYNTAX)

Identifier	::=	Letter (Letter|Digit)*
Integer-Literal	::=	Digit Digit*
Character-Literal	::=	**'** Graphic **'**
Operator	::=	**+** | **-** | ***** | **/** | **<** | **>** | **=**
Letter	::=	a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
Digit	::=	0|1|2|3|4|5|6|7|8|9
Graphic	::=	any single ascii character

NOTES on reduced triangle.

The EBNF here defines the language you are to scan and parse. It is very similar to the language you have seen so far in the lab sessions with some extra functionality.

Tokens highlighted in **red** are reserved characters in the language and should be treated as separate tokens. These token kinds already exist in the TokenKind class you have been given.

The terminals or Microsyntax here also already exist as token in your TokenKind class.

Remember that EBNF uses regular expressions to define optional and repeated statements so

Command ::= single-Command (;single-Command)* means that a Command is composed of one single-Command followed by 0 or more single-Commands

primary-Expression ::= Identifier (actual-parameter-sequence) means that a primary-Expression is composed of an Identifier which may be followed by a parameter list