UNIVERSITY OF NAIROBI

FACULTY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTING AND INFORMATICS

CSC 326: COMPILER CONSTRUCTION

MINI LANGUAGE CONSTRUCTION

Done by Group 17:

P15/137032/2019	Hamud Abdulrahman Abeid
P15/1894/2020	Emmanuel Mbithi Muthiani
P15/1915/2020	Mugo Staicy Nelima Muthoni

CONTEXT FREE GRAMMAR IN BNF FORMAT

```
<program> ::= <statement> <program'>

<program'> ::= <program> | ε

<statement> ::= <assignment> | <loop> | <conditional> | <print>

<assignment> ::= <identifier> = <expression> <EOL>

<loop> ::= while ( <condition> ) { <program> }

<conditional> ::= if ( <condition> ) { <program> } <else>

<condition> ::= <expression> <op> <expression>

<else> ::= else { <program> } | ε

<print> ::= print ( <expression> ) <EOL>


<expression> ::= <term> <expression'>

<expression'> ::= <add-op> <term> <expression'> | ε

<term> ::= <factor> <term'>

<term'> ::= <mult-op> <factor> <term'> | ε

<factor> ::= <identifier> | <number> | ( <expression> )


<op> ::= <rel-op> | <and-op> | <or-op>

<add-op> := + | -

<mult-op> ::= * | /

<rel-op> ::= < | > | <= | >= | == | !=

<and-op> ::= &&

<or-op> ::= ||


<keyword> ::= if | else | while | print


<identifier> ::= <letter> <identifier'>

<identifier'> ::= <letter-or-digit> <identifier'> | ε

<letter> ::= a | b | c | ... | z | A | B | C | ... | Z | _

<digit> ::= 0 | 1 | 2 | ... | 9

<letter-or-digit> ::= <letter> | <digit>

<number> ::= <digit> <number'>

<number'> ::= <digit> <number'> | ε
```

```
<EOL> ::= ;
```

SYNTAX FOR THE CONTEXT FREE GRAMMAR

<program> represents a program and consists of one or more statements followed by <program'>.

<program'> represents the continuation of a program and can be either an end-of-line (<EOL>) followed by another statement, or it can be empty (ε) to indicate the end of the program.

<statement> represents a single statement and can be an assignment, loop, conditional, or print statement.

<assignment> represents an assignment statement and consists of an identifier, an equals sign, an expression, and an end-of-line (<EOL>).

<loop> represents a while loop and consists of a condition expression enclosed in parentheses, followed by a program enclosed in curly braces ({}).

<conditional> represents an if statement and consists of a condition expression enclosed in parentheses, followed by a program enclosed in curly braces. It can also have an optional else clause, represented by <else>.

<condition> represents logical expressions in if and while statements

<else> represents the else clause of an if statement and can either be a program enclosed in curly braces, or it can be empty (ε) to indicate that there is no else clause.

<print> represents a print statement and consists of the keyword print, an expression enclosed in parentheses, and an end-of-line.

<expression> represents an expression and consists of one or more terms followed by <expression'>.

<expression'> represents the continuation of an expression and can be either an addition or subtraction operator (<add-op>) followed by another term and another <expression'>, or it can be empty (ε) to indicate the end of the expression.

<term> represents a term and consists of one or more factors followed by <term'>.

<term'> represents the continuation of a term and can be either a multiplication or division operator (<mult-op>) followed by another factor and another <term'>, or it can be empty (ε) to indicate the end of the term.

<factor> represents a single term, which can be an identifier, a number, or an expression enclosed in parentheses.

<op> represents logical operators

<add-op> represents an addition or subtraction operator.

<mult-op> represents a multiplication or division operator.

<rel-op> represents a relational operator, such as less than (<) or greater than (>).

<and-op> represents a logical and operator (&&).

<or-op> represents a logical or operator (||).

<keyword> represents a keyword, such as if, else, while, or print.

<identifier> represents an identifier, which is a string of one or more letters, digits, or underscores (_), starting with a letter or underscore.

<identifier'> represents the continuation of an identifier, which can be any number of letters, digits, or underscores.

<letter> represents a single letter, either lowercase or uppercase, or an underscore.

<letter-or-digit> represents a single letter or digit.

<number> represents a number, which is a string of one or more digits, starting with a digit.

<number'> represents the continuation of a number, which can be any number of digits.

<EOL> represents an end-of-line, which is a semicolon (;).

TOKENS AND LEXEMES USED.

The lexemes in the CFG are:

Keywords: if, else, while, print

Operators: + , - , * , / , < , > , <= , >= , == , != , && , ||

Symbols: = (,) , { , } , ;

Identifiers: any string of letters, digits, or underscores starting with a letter or underscore, such as x, x4, y_1, money etc.

Numbers: any string of digits, such as 123, 45, 0, -678