**Lexical Conventions**

**Comments**

A comment, whether single or mult-line, goes between <- characters, which indicates the start of it and ->, which indicate the end. The comment can be placed anywhere in the program as long as it is between these two characters.

Ex:

1. <- create a chord with three notes lengthened by 1/8 ->
2. Note c = (C, 0, *half*);
3. Note g = (G, 0, *half*);
4. Note e = (E, 0, *half*);
5. Chord cr = (c, g, e, eight)

**Identifiers**

In Cb language, an identifier, is a sequence of letters, digits, and underscores (\_).

Note that an identifier must always starts with a lower case letter. There is no limit on how long an identifiers can be.

Below is the list of characters allowed in creating an identifier.

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 \_

0 1 2 3 4 5 6 7 8 9

**Ex:** nice\_note, nICE\_note, and n\_NOTE2 are acceptable identifiers. However,

\_nicenote, Nice\_note, and 2nicenote are not acceptable identifiers

**Keywords**

They are identifiers used to specify the types of expressions, for retrieving/including methods from an external packages. These keywords listed below are reserved for Cb, which means that they cannot be used as normal identifiers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Int | use | Use | sixteen |  |  |
| Note | if | method | eight |  |  |
| Chord | else | return | quarter |  |  |
| Scale | while | compose | half |  |  |
| Stanza | foreach |  | whole |  |  |
| Score | in |  | b |  |  |
| String |  |  |  |  |  |

**Literals**

Cb uses only Integer literals that consist of a sequence one or more digits.

**Constants**

**Integer constant.**

Cb has a set of Integer constants that are used to represent basic notes and known durations of notes. Below is a list of Integer constants:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G |
| A# | B# | C# | D# | E# | F# | G# |
| Ab | Bb | Cb | Db | Eb | Fb | Gb |

**Operators**

An operator specifies an operation to be performed. Some of operators used in Cb language are shown below:

Cb language takes advantage of existing arithmetic operators to manipulate notes and chords. Below is a list of Operators:

++ -- \* + - / ^- ^+ % # b

< > <= >= is isnt and or

**=**

**Punctuators**

A punctuator is a symbol that add a semantic value to the expression or statement that it belongs to, but does not perform an operation. These punctuators are used in declaration and assignment of variables. Below is a list of Punctuators:

[ ] ( ) { } . ;

Ex:

1. **Note asharp = (A, 0, *quarter*);**
2. <- do re mi song ->

method Stanzas doremi(Int duration){

Note do = (C, 0, duration);

Note re = (D, 0, duration);

Note mi = (E, 0, duration);

Stanza s = [do, re, mi];

return s;

}