**LANGUAGE OVERVIEW**

LexiCom Compiler Programming Language is derived from C, VB.net and Swift Language's where the advantages involve its concise and simple syntax.

**GENERAL RULES**

1. The program must have a *Lead* function. The word *Lead* is followed by a colon “**:**”.
2. The *Lead* function must begin with the word *Start* and end with the word *End.*
3. *LexiCom* is white space insensitive.
4. *LexiCom* is a case sensitive Programming Language. Thus, *Automata*  and *AuToMaTa* are two different identifiers.
5. Declarations can be made locally or globally.
6. There is no specific arrangement in declaring.
7. *Var* is used for declaring a variable and *Let* for constants.
8. *Task* definitions are placed after the *Lead* function. To use these tasks they must first be declared before the *Lead* function.
9. Reserved words must not be used as an identifier.
10. Identifiers must start with a capital letter. It must not begin with a special characters or a small letter.
11. Identifiers must have a minimum of 1 character, and a maximum of 12 characters
12. Underscores “**\_**” are allowed in naming an Identifier as long as it is not the first character.
13. Statements are terminated with a period “**.**”.
14. The whole program must be terminated by the word *End* followed by the terminator period “**.**” then the number sign “#”.
15. Statements after the number sign “#” will not be read.

**STRUCTURE OF THE PROGRAM**

Before we introduce the basic building blocks of LexiCom programming language, let us look a bare minimum LexiCom program structure so that we can fully understand the Language in the next pages

LexiCom Hello World Example

A LexiCom program basically consists of the following parts:

* Declarations:
  + Task Declarations
  + Object Declarations
  + Global Variable Declarations
* Lead Function
* End of the program
* Task Definition

|  |
| --- |
| ***[<Task declaration>]***  ***[<Object declaration>] [<Global variable declaration>]***  ***Lead: Start  [Statement\_1.]***  ***[Statement\_2.]  …***  ***[Statement\_n.] End. [<Task definition>]#*** |

Let us look at a simple code that would print the words *“Hello World”:*

**Lead:**

**Start**

**Say “Hello World!”.**

**End.#**

Let us look at the various parts of the program:

1. The first line of the program, *Lead:* is the main function where the program execution begins.
2. The next line *Start,* it serves as the open curly brace “{” in C.
3. The next line *Say “…”* which displays *“Hello World!”* on the screen while execution of the program.
4. The last line *End.#* terminates the whole program.