Language Reference

This language manual is written and described by the **team members of Akatsuki**.

- **Indentation :**

We are following a proper indentation rule. Like if we are writing any function so the compiler will only take those statements inside it which are properly indented and once the indentation is broken the compiler will see it as the end of that function. Like if we are following a indentation rule of 4 spaces so in that whole function everything should be at least 4 space indented or the compiler will see it as another function or a function inside a function.

- **Variable declarations :**

No need to specifically declare a variable. We just have to write the variable name and it's value; the compiler will automatically see it's type and declare it itself only. In C we had to specifically declare a variable and it’s type. Here the language is having fluidity so it’s easy here.

- **Input and output function :**

  Again we can see here too that in C it was very difficult for a new programmer to remember all the symbols perfectly and execute it. Here the input function is “in” and the output function is “out” and whatever we are gonna input or output should be in brackets with quotes.

- **English Keywords :**

  We are using english keywords like “And” , “or” , “not” like keywords instead of using symbols “&&” , “||” , “!=” respectively as these symbols are hard to use, difficult to remember and also takes more time for any programmer to write. So we brought the concept of writing what we are saying in our brain. So every time we are writing any multi conditional statement we are just typing the words and not some freaky symbols.

- **“For” loop :**

  As we were seeing in the C language the syntax of the for loop is a bit tricky with a lot of semicolons and brackets. In this language we took the syntax to the very peak of simplicity. In the C language the syntax of for loop will be like

  For (int i = 0; i<n ; i++), as you can see here that syntax is a bit tricky but in our language the syntax is like,

  For i in (n) , here you can see that the syntax is completely simple also for specially declaring the start of the variable and also the increment we can just write it like this,

  For i in (1,n,2) here i will start from 1 and will be incremented by 2 every time.

- **Array :**

We have changed the syntax of the array too. Now we don’t have to declare the type of value it’s going to store. We will just have to declare the array and the type of data will be seen by the compiler automatically. The syntax now is “array dec[5]”

- **Comments :**

  We are going to use the conventional method to write the comments. For single line comments we are going to use “//” and for multi lines it will be “$.....This is our new language….$”

- **Keywords :**

The keywords that we used till now are if, else, for, return, for, var, import, while etc.. in our examples.

We were programming our programs in C language and we came across multiple problems like the complexity of syntax. Everytime when we are writing any code we have to deal with a lot of stress to make the syntax proper. Sometimes it is even happening that our code is giving a critical error just because of one small syntax error of semi colon or some brackets. So, we took these things out completely . Now our language is not having any semicolon and lots of brackets because it is indentation based as lots of modern languages are right now e.g. Python. Now sometimes there are some difficult syntax that we have to memorize and pay a lot of effort like in for loops and remembering the syntax in multi conditional statements. So we tried to write the same as we are speaking, like replacing && with and directly. Also in the for loops we simplified the syntax a lot and now it's just a child's play to write it. Now we just have to write it like "for i in (1,20,1)" which is less time consuming.

Now there is no need to declare the type of any variable, which is again a confusion, now we just have to declare the variable and we are good to go.