

WELCOME TO RYON

A programming language that will be somehow similar to Python will use C# compiler. Although the keywords, structure will be different from others. It will be dynamic in nature. The main features of language will be variables, loops, conditional statements and methods. The main purpose of this language is to provide an easy way for the young developers to learn the concepts of programming and start to get their hands dirty. The language will be case-sensitive.

DECLARATION OF VARIABLES:-

The variable data type will be inferred by the compiler. Variable names will contain alphabets and digits. The variable names must start with ~. Once the data type is being inferred by the compiler its data type won't change in the whole program. Apart from that ~ must not contain anywhere in the variable name.

RE: (~A)[(A + D)]*

For example: ~name1

DECLARATION OF CONDITIONAL STATEMENTS:-

The conditional statement will have IF and ELSE as keywords for them. The condition after IF and ELSE must be enclosed in parenthesis. The thing that must return from the parenthesis can be 0, 1, any character, any digit, True or False.

Syntax:

If (condition)

{

//Statements

}

Else

{

//Statements

}

DECLARATION OF LOOPS:-

The loops this language will have is for, while, do while, foreach. But the syntax will be changed from the traditional syntaxes.

Syntax:

For loop:

//Same as C#

While loop:

For-condition (condition)

```
{  
//Statements  
}
```

While (True):

```
For-True ()  
{  
//Statements  
}
```

Do-While (condition):

```
For-first( condition )  
{  
//Statements  
}
```

OPERATORS:-

//Same as C#.

Power operator ^.

METHODS:-

Methods will be defined by the syntax:

```
Method <return_type> <method_name>( Parameters )  
{  
//Statements  
}
```

The datatype must be included in the definition of the datatype.

TYPES OF VARIABLES:-

The variables defined inside a method will have a local scope.

The variable can be constant.

The variable can be first constant meaning a variable whose value can only be changed once.

ARRAY:-

In our language, we will support only 1D firstly.

The syntax of the array will be.

```
Array[ <no_of_elements> * <data_type> ] <variable_name>;
```

Or

```
Array[] <variable_name> = [ elements ]
```

In the second type the data type and the length of the array will be automatically inferred by the compiler.

DATA TYPES:-

In our language we will support multiple data types:

1. Integer – this will control all the integers irrespective of the size.
2. Float – this will deal with all the decimal point values.
3. String – will deal with the array of characters.
4. Char – any single character.
5. Object – the main datatype. All the other datatypes inherit from Object datatype.

LISTS:-

It is same as array with the exception that the number of items won't be specific.

The syntax of defining such list is:

```
VarArray[ <no_of_elements> * <data_type> ] <variable_name>;
```

Or

```
VarArray[] <variable_name> = [ elements ]
```

Of course there will be some predefined set of functions available to each but we will discuss that later.

SET:-

It is a type of list with the exception that it will always keep the list sorted and no multiple entries of single variable will be allowed.

The syntax of it will be:

```
Set[ <no_of_elements> * <data_type> ] <variable_name>;
```

Or

```
Set[] <variable_name> = [ elements ]
```

KEYWORDS:-

The keywords can be included as the language will evolve.

If

Else

For

For-True

For-First

Integer

Float

String

Array

VarArray

Set

Method

return

for-condition

STRINGS:-

In our language strings will be immutable. You can't edit a string if that string is once assigned. They can be reassigned. Adding two strings will result in the formation of a new string. A string can be indexed through square brackets. The start of the string will occur from the 0th position. If two strings are concatenated and the resultant string is stored in one of the concatenated strings than the previous value of that string variable will be collected by the garbage collector and the variable will now assign to the newly assigned value.

The only default method that our language will give to the user will be just Length() function. The function will return an integer representing the total characters that string have in it.

INTEGERS:-

Integers in this language can of any length. The compiler will handle all the integers' manipulation. By default, integers will have long datatype. The only functions will be providing with integers are the Floor(), Ceil() and Round() functions. The integers will be saved on stack.

FINAL WORDINGS:-

The main idea behind this is that we want to create a language that is will be easy to use. But in real it will help the young programmers to get started with programming. The main idea behind not providing

any functions is that we want the programmer to get idea and logic of each concept of how to do certain tasks. It will not only help them to think like a computer but also will make them feel comfortable when they will switch to another language.