Language Specification

# **Objective**

“Programming teaches us How to think”

**By Steve jobs**

Our Goal is to make programming language which can easily be understood by Secondary School Students and it can help them to improve their problem solving skills. Most importantly students can experience what Programming is? and How ideas can be converted into a real product ? That’s why name of language we call it Y++ which means y=y+1 \\ youToday=youYesturday + 1

Each and Everyday Programmer can improve his/her skills by using Y++

# **About Y++**

Y++ will not be Object Oriented language because goal is to make language simple so students at school level can easily understand. Y++ will be a Procedural language just as C language but with Simple Syntax.

# **Regular Expression**

Names of Variables and functions can start only from letters and end with any combination of letter or digits

Let

L = a, b, c, d, e, f, …z A, B, C, D, E, F, …Z

D= 0, 1, 2, 3, 4, 5 … 9

RE-> (L) (L+D)\*

# **DATA TYPES**

|  |  |
| --- | --- |
| **C-Language** | **Y++** |
|  |  |
| Int | Numbers |
| Float | Decimals |
| string | Words |

# **ESCAPE SEQUENCE**

|  |  |
| --- | --- |
| **Escape Sequence** | **Represents** |
|  |  |
| \n | New Line |
| \t | Horizontal Tab |
| \’ | Single Quotation |
| \’’ | Double Quotation |

# **INPUT/OUTPUT**

|  |  |
| --- | --- |
| **C/C++** | **Y++** |
|  |  |
| cout<<”Programming is for Everyone”; | show(“Programming is for Everyone”); |
| cout<<”Num is ”<<num; | show(“Num is”@num); |
|  |  |
| cin<<num; | putin(num); |
| cin<<x1<<x2; | putin(x1@x2); |

# **CONDITIONAL STATEMENTS**

IF/Else Statements is same in Y++ as in C language

|  |  |
| --- | --- |
| **C-Language** | **Y++** |
| Int num=10;  Switch(num)  {  case num%2 == 0 :  cout<<”Num is Even”<<num;  break;  case num%2 == 1 :  cout<<”Num is ODD”<<num;  break;  } | Numbers num=10;  Check(num)  {  expr num%2 == 0 :  show(“Num is Even”@num);  stop;  expr num%2 == 1 :  show(“Num is ODD”@num);  stop;  } |

# **LOOP STATEMENTS**

While loop

|  |  |
| --- | --- |
| **C-Language** | **Y++** |
| While( counter<1000)  {  Cout<<”Sorry Teacher I will not do it again \n”;  counter++  } | RepeatTill(1000)  {  show(“Sorry Teacher I will not do it again \n”);  } |

# Here Simply Specify integer number for repetition

Repeat(Numbers num)

{ # Statements

}

For loop

|  |  |
| --- | --- |
| **C-Language** | **Y++** |
| For(int i=0; i<10; i++)  {  //Statements  } | Repeat(Numbers i=0; i<10; i++)  {  } |

# **PREPROECESSOR DIRECTIVES**

|  |  |
| --- | --- |
| **C-Language** | **Y++** |
| #include<stdio.h> | Addlib<stdio.h> |

# **COMMENTS**

|  |  |
| --- | --- |
| **Y++** | **Represents** |
| # | Single Line |
| ## | Multiple Lines |

# **FUNCTIONS**

**Main Function**

|  |  |
| --- | --- |
| **C-Language** | **Y++** |
| Void main()  {  // Statements  }  int main()  {  //Statements  return 0;  } | start()  {  #Statements  }  Numbers start()  {  #Statements  send 0;  } |

**Function Declaration/Defination/ Calling**

**returnType** functionName (**dataType** parameter1Name , … , **dataType** parameter2Name); #Prototype

**returnType** functionName (**dataType** parameter1Name , … , **dataType** parameter2Name)

{ statement 1; statement 2; statement 3; … statement x; **send** returnValue; } #Defination

functionName(parameter1Name); # Function Calling

**Built-in Functions**

There will be math library for regular Arthematic operation Such as sum, sub, div, product, power…etc

Addlib<libraryName>