RESERVED WORDS

|  |  |  |  |
| --- | --- | --- | --- |
| C Languange | C++ Language | Proposed Language | Description |
| Main | Main | !Start | The Function that will run when the user starts the program |
| Data Types | | | |
| String | String | String | A data type that holds multiple characters. |
| Bool | Bool | Boolean | A data type that can only hold the value of ‘true’ or ‘false’. |
| int | int | Number | A data type that that can store Integers. |
| float | float | Decimal | A data type that will hold a value of numbers with decimals. |
| void | void | Null | A return type that will not return anything. |
| Data Structures | | | |
| struct | struct | File | Collections of variables placed under a single name. |
| I/O Statements | | | |
| printf | cout | Write | Used to print characters. |
| scanf | cin | Read | Reads data that has been inputted by the user. |
| Conditional Statements | | | |
| if | if | If | Executes a block of code if a specified condition is true |
| else | else | Else | Specify a block of codes to be executed, if the condition is false. |
| else if | else if | ElseIf | Specify a new condition of test, if the first condition is false. |
| switch | switch | Switch | Select one of many blocks of code to be executed. |
| case | case | Choice | Choices of conditions on the switch. |
| default | default | Default | Specify a block of codes, if all the Choices are wrong. |
| break | break | Stop | Stop the execution |
| continue | continue | Jump | Jump to the next statement |
| Looping Statements | | | |
| for | for | For | Executes a block of codes until the condition is true. |
| do | do | Do | Executes a block of code, once. |
| while | while | While | Executes a block of code until the condition is true. |
| Other Statements | | | |
| true | true | True | Boolean value |
| false | false | False | Boolean value |
| return | return | Return | Returns the value to the caller function. |
| - | - | Start | Marks the start of a function. |
| - | - | End | Marks the end of the function. |
| - | - | End! | Marks the end of !Start (Main) Function |
| && | && | And | If both operands are true, the condition will be true. |
| || | || | Or | If any operand is true, the condition will be true. |
| - | - | Fixed | Used to declare constant |
| \n | \n | Newline | Used to have a new line on the program. |
| \t | \t | Tab | Used to have a tab on the program. |

RESERVED SYMBOLS

*1. Arithmetic operators*

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + | Adds two operands | 10 + 5 will give 15 |
| - | Subtracts second operand from the first | 10 – 5 will give 5 |
| \* | Multiplies both operands | 10 \* 5 will give 50 |
| / | Divides numerator by de-numerator | 10/5 will give 2 |
| % | Modulus Operator and remainder of after an integer division | 10 % 5 will give 0 |
| ++ | increases integer value by one | 10++ will give 11 |
| -- | decreases integer value by one | 10—will give 9 |

*2. UNARY OPERATORS*

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| ++ | increases integer value by one | 10++ will give 11 |
| -- | decreases integer value by one | 10—will give 9 |

*3. RELATIONAL OPERATORS*

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| == | Checks if the values of two operands are equal or not, if yes then condition becomes true. | (10 == 5) is false |
| != | Checks if the values of two operands are equal or not, if values are not equal then condition becomes true. | (10 != 5) is true |
| > | Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true. | (10 > 5) is true |
| < | Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true. | (10 < 5) is not true |
| >= | Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true. | (10 > 5) is true |
| <= | Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true. | (10 < 5) is not true |

*4. LOGICAL OPERATOR*

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| ! | Used to reverse the current state of the operand. | !(10 == 10 && 5 == 10) is true |

*5.* *ASSIGNMENT OPERATOR*

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| = | Simple assignment operator, Assigns values from right side operands to left side operand | A = 10 + 5 will assign value of 10 + 5 into A |
| += | Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand | A += 10 is equivalent to A = A + 10 |
| -= | Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand | A -= A is equivalent to A = A - 10 |
| \*= | Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand | A \*= A is equivalent to A = A \* 10 |
| /= | Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand | A /= 10 is equivalent to A = A / 10 |

*6. OTHERS*

|  |  |
| --- | --- |
| **Operator** | **Description** |
| . | Terminator |
| , | Separator |
| “” | Used in defining the value of a String |
| ( ) | Used in different operations |
| [ ] | Used in declaring Arrays |
| @ | Used to access an element of a file |