**TASK\_01:**

**Download Language Specification Document Template from google class room and complete**

**the document for C++ Language.**

**Note: Template for Language Specification Document is available at google class room.**

**Motivation:**

**C++** is a procedural and a general-purpose object-oriented **programming language developed** for implementing OOP concepts which are not introduced in C language and also the syntax is more user friendly for users.

**Targeted Audience:**

Programmers, young generation

**Language Paradigm:**

Multi-paradigm (procedural, object oriented);

**Case sensitivity:**

It is case sensitive.

**Keywords:**

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Keyword** | **Keyword** |
| for | main | Try catch |
| double | while | throw |
| int | case | public |
| else | switch | void |
| do | float | namespace |
| this | continue | private |
| string | foreach | boolean |
| break | default | protected |

**Data Types:**

* Integer
* Double
* Character
* Boolean
* void
* Float
* String
* Short integer
* Long integer
* Short double
* Long double

**Iterative statements:**

* For loop
* foreach
* While loop
* Do while loop

**Conditional Statements:**

* if
* else if
* switch

**Comments (Multi line + Single line):**

* //
* /\* \*/

**Line terminator:**

* ;

**Operators (Mention Each Operator with their class part):**

* +
* -
* /
* %
* \*
* &&
* ||
* !=
* ==
* =
* >=
* <=

**Punctuators (Mention each punctuator with their class part):**

* []
* {}
* ,
* .
* ;
* ->
* ()
* “”

**Identifiers:**

* Identifiers always starts with “\_” or alphabets.
* There is no space in identifiers.
* Identifiers can’t start with any number or special character.

**Syntax Specification:**

**Start:**

Using namespace std

Int main(){

}

**End**:

Int main(){

}//end

**Declaration :**

Int a=3;

Double i=0.2;

**Functions:**

Void name(){  
}

Void name(int a){  
}

Int name(int a){

Return a;

}

**Array 1D/2D:**

Int arr[size]={1,2,3};

Int arr[row][column]={1,2,3

1,3,4};

**Classes/structures:**

#include <iostrream>

using namespace std;

class name

{

    // Access specifier

    public:

    // Data Members

    string name;

    // Member Functions()

    void changename()

    {

       cout << "name is: " << name;

    }

};

Struct name{

Int id;

Double name;

}

**Loops :**

For(int i; i<num.length; i++){

}

for(int i; i<num.length; i++){

for(int j; j<num.length; j++){

}

}

Int i=0;

While(i<=number){

//code

i++;

}

Do{

//code

}while(Condition);

**Conditional statements:**

if(num>0){

}

Else if(num<0){

}

else{

}

* Switch(expression){

Case constant-expression:

Statement(s);

Break;

Case constant-expression:

Statement(s);

Break;

Default:

Statement(s);

}

**TASK\_02:**

**Language: Set of strings of a’s and b’s ending with the string abb. So L = {abb, aabb, babb,**

**aaabb, ababb, …………..}**

**1. Generate RE and DFA of above Language.**

**2. Implement DFA generate in a).**

#include<iostream>

#include<string.h>

int trans(int s,char c);

using namespace std;

int main(){

int cs = 0;

int fs = 3;

string s;

int i=0;

cout<<"Input any string : ";

cin >> s;

while(s[i]!= NULL)

{

cs = trans(cs,s[i]);

i++;

}

if(cs==fs)

{

cout<<"valid string"<<endl;

}

else{

cout<<"invalid string"<<endl;

}

}

int trans(int s ,char c)

{

int TT[4][2] = {

{1,0},{1,2},{1,3},{1,0}

};

if(c=='a')

{

return TT[s][0];

}

else if(c=='b')

{

return TT[s][1];

}

else

{

return -1;

}

}