C++ Documentation

Variables

- Int ->(Stores an integer value)
- float -> (Stores decimal point values)
- double -> (similar to float but it has more precision)
- long -> (Used for storing large integers)
- char -> (Used to store single characters)
- bool -> (Stores Boolean value)
- string -> (Stores a sequence of characters)

Examples:

int a = 5;

float b = 6.98;

double c = 8.976543;

long d =6748382119427;

char e = ‘A’ (Single inverted commas are used)

bool flag =true or flag=false;

string s = “CSI”;

BASIC PRINTING STATEMENTS AND SCANNING STATEMENTS

cout<<”Hello C++”<<endl;

(this will print Hello C++ in terminal)

(endl is used to give new line)

int n;

cin>> n;

(This will scan the value given by user in keyboard and will store it in variable n)

LOOPS IN C++

Major two loops in c++

1] While loop

2] For Loop

Syntax of while loop:

```
while(condition)
{
        statements;
        increment/decrement;
}
```

Eg:

Program to print numbers from 1 to 5

```
int i=1;
while(i<=5)
{
        cout<<i<<endl;
        i++;
}
```

FOR LOOP

Syntax:

```
for(initialization;condition;increment/decrement)
{
        Statements
}
```

Example:

```
for(int i=1;i<=5;i++)
{
        cout<<i<<endl;
}
```

ARRAYS AND VECTORS IN C++

ARRAY SYNTAX

Datatype variable_name[size];

Eg:

int a[4];

(This will create an array of name a and a size of 4)

(Array starts with 0 based indexing so it will have start from 0 to 3)

To store values:

a[0]=1;

a[1]=3;

or we can use for loop to store in array.

2D ARRAY

Eg : int a[4][4] will create a matrix of size 4 by 4

(The size of array will be the number specified to it, we cannot increase it, to overcome this we use Vector)

VECTOR SYNTAX:

vector<datatype> name;

Eg: vector<int> v;

To insert element we do:

v.push_back(1); // This will add 1 into vector

v.pop_back(); //This will remove the last element in vector;

We can all use v[0] , etc to access vector elements.

FUNCTIONS IN C++

SYNTAX:

```
Datatype function_name(arguments)
{
        Statements
        Return value
}
```

Example:

```
int fun(int n,string s)
{
        cout<<s;
        return n;
}
```

(Note : functions should return those value as of their type: Eg: fun is function of int type so we should return an integer value, if we have nothing to return we can use void type in functions)