**C++ Access Specifier Public Private Protected**

c++ access specifier public private protected. Access modifiers defines the scope of methods or variables. Scope means whether the methods or variable defined in class can be accessed by all class of project or the same class only or with few other class which is inherited.

C++ defines three access modifiers or access specifiers:  
**1. Private:** Private member defines that the members or methods can be accessed within the same class only.

**2. Public:** Public member defines that the variable or methods can be accessed at any where within  
the project.

**3. Protected:** Protected member can be accessed to the class which is inherited by other class.

By default, all members and function of a class is private i.e if no access specifier is specified.

Syntax of declaring access modifiers in c++

class

{

private:

// private members and function

public:

// public members and function

protected:

// protected members and function

};

Here is an example which uses all the three access specifier in the same class.

#include<iostream>

#include<conio.h>

using namespace std;

class Declare

{

private:

int a = 10;

public:

int b = 20;

protected:

int c = 30;

public:

void call()

{

cout<<"Declare variable values"<<endl;

//Every members can be access as all are in the same class

cout<<"Value of a = "<<a<<endl;

cout<<"Value of b = "<<b<<endl;

cout<<"Value of c = "<<c<<endl;

}

};

class Inherit:public Declare

{

public:

void call()

{

cout<<"\nInherit variable values"<<endl;

//As 'a' is private member so 'a' cannot be accessed by another class

//cout<<"Value of a = "<<a<<endl;

//'b' is declared as public, so it can be accessed from any class which is inherited

cout<<"Value of b = "<<b<<endl;

//'c' is declared as protected, so it can be accessed from class which is inherited

cout<<"Value of c = "<<c<<endl;

}

};

int main()

{

Declare d;

d.call();

Inherit i; //= new Inherit();

i.call();

cout<<"\nAccessing variable of declare outside declare class"<<endl;

//'a' cannot be accessed as it is private

//cout<<"value of a = "<<d.a<<endl;

//'b' is public as can be accessed from any where

cout<<"value of b = "<<d.b<<endl;

//'c' is protected and cannot be accesed here

//cout<<"value of c = "<<d.c<<endl;

}

