**Friend**

One of the important concept in OOP is data hiding, But sometimes we may need to write long and complex code so to avoid it we have friend function / friend class.

> Friend function in C++

> Friend class in C++

Friend Function:

if a function is defined as friend function then, the private and protected data of a class can be accessed using the friend function.

Syntax:

*class class\_name*

*{*

*... ... ...*

*friend return\_type function\_name(arguments);*

*... ... ...*

*}*

Here,

The friend function can be defined as normal function to access data of the class, No friend keyword is used in the definition.

Note:

friend funcion is not a member function of the class but it can modify the data members in the class, friend function can be invoked normally.

Example:

*class B;*

*class A*

*{*

*private:*

*int numA;*

*public:*

*A():numA(10){}*

*friend int add(A, B);*

*};*

*class B*

*{*

*private:*

*int numB;*

*public:*

*B(): numB(11){}*

*friend int add(A, B);*

*};*

*int add(A objA, B objB)*

*{*

*return (objA.numA + objB.numB);*

*}*

*int main()*

*{*

*class B;*

*class A*

*{*

*private:*

*int numA;*

*public:*

*A():numA(10){}*

*friend int add(A, B);*

*};*

*class B*

*{*

*private:*

*int numB;*

*public:*

*B(): numB(11){}*

*friend int add(A, B);*

*};*

*int add(A objA, B objB)*

*{*

*return (objA.numA + objB.numB);*

*}*

*int main()*

*{*

*A Oa;*

*B Ob;*

*std::cout << "Sum :" << add(Oa, Ob) << std::endl;*

*return 0;*

*}*

Friend Class:

A class can be made friend of another class using the keyword *friend*.

Syntax

*class B;*

*class A*

*{*

*// class B is friend of class A*

*friend class B;*

*... ... ...*

*}*

*class B*

*{*

*... ... ...*

*}*

Here,

All the member function of class B will be friend function of class A. So member function of class B can access private and protected members of class A. But member function of class A cannot access data of class B.