9.Friend function

#include<iostream>

using namespace std**;**

class myclass

**{**

private**:**

int a**,**b**;**

public**:**

friend int sum**(**myclass x**);**

myclass**(**int i**,**int j**);**

**};**

myclass**::**myclass**(**int i**,**int j**)**

**{**

a**=**i**;**b**=**j**;**

**}**

int sum**(**myclass x**)**

**{**

**return** x**.**a**+**x**.**b**;**

**}**

int main**()**

**{**

myclass n**(**3**,**4**);**

cout**<<**sum**(**n**);**

**}**

10.Friend function

//friend return type\_name(class\_name object)

//friend keyword can be friend of more class

#include<iostream>

using namespace std**;**

class A

**{**

private**:**

int a**,**b**;**

public**:**

void input**()**

**{**

cout**<<**"Enter 2 value for a and b : "**;**

cin**>>**a**>>**b**;**

**}**

friend void add**(**A ob**);**

**};**

void add**(**A ob**)**

**{**

int c**;**

c**=**ob**.**a**+**ob**.**b**;**

cout**<<**"sum="**<<**c**;**

**}**

int main**()**

**{**

A obj**;**

obj**.**input**();**

add**(**obj**);**

**}**

11.Friend function

#include <iostream>

using namespace std**;**

class A **{**

private**:**int a**;**

public**:**

A**()** **{** a **=** 0**;** **}**

friend class B**;** // Friend Class

**};**

class B **{**

public**:**

void showA**(**A x**)**

**{**

// Since B is friend of A, it can access

//private members of A

cout **<<** "A::a=" **<<** x**.**a**;**

**}**

**};**

int main**()**

**{**

A a**;**

B b**;**

b**.**showA**(**a**);**

**return** 0**;**

**}**

12.Friend function

//friend return type\_name(class ab)

//friend keyword can be friend of more class

#include<iostream>

using namespace std**;**

class A

**{**

int a**,**b**;**

public**:**

void input**()**

**{**

cout**<<**"Enter 2 value for a and b : "**;**

cin**>>**a**>>**b**;**

**}**

friend void add**(**A ob**);**

**};**

void add**(**A ob**)**

**{**

int c**;**

c**=**ob**.**a**+**ob**.**b**;**

cout**<<**"sum="**<<**c**;**

**}**

int main**()**

**{**

A kk**;**

kk**.**input**();**

add**(**kk**);**

**}**

12.Friend function

#include <iostream>

using namespace std;

class B;

class A{

public:

void showB(B);

};

class B

{

private: int b;

public:

B() { b = 0; }

friend void A::showB(B x); // Friend function

};

void A::showB(B x)

{

// Since showB() is friend of B, it can

// access private members of B

cout << "B::b = " << x.b;

}

int main()

{

A a;

B x;

a.showB(x);

return 0;

}