**Pointer to Members of the class**

**Reference Link:** <http://www.informit.com/guides/content.aspx?g=cplusplus&seqNum=142>

### **Pointers to Member Functions and Pointers to Data Members**

A class can have two general categories of members: functions and data members. Similarly, there are two categories of pointers to members: pointers to member functions, and pointers to data members. The latter are less common because you rarely have direct access to data members. However, when using legacy C code with structs or classes have public data members, pointers to data members might be useful.

### Declaring Pointers to Data Members

The syntax of pointers to members might look confusing at first, but it's consistent. The syntax resembles the form of ordinary pointers to functions, with the addition of the class name followed by the operator ::. For example, if an ordinary pointer to int looks like this:

int \* pi;

you define a pointer to an int member of class A like this:

class A{/\*\*/};

int A::\*pmi; // pmi is a pointer to an int member of A

You can initialize pmi like this:

class A

{

public:

int num;

int x;

};

int A::\*pmi = &A::num; // 1

#### Manipulating a data member through an object

The statement numbered 1 defines a pointer to an int member of class A and initializes it with the address of num. Now you can use the pointer pmi to examine and modify num's value in any object of class A:

A a1;

A a2;

int n = a1.\*pmi; // copy the value of a1.num to n

a1.\*pmi = 5; // assign the value to a1.num

a2.\*pmi = 6; // assign the value 6 to a2.num

#### Manipulating a data member through an object's pointer

Similarly, you can access a data member through a pointer to A like this:

A \* pa = new A;

int n = pa->\*pmi; // assign to n the value of pa->num

pa->\*pmi = 5; // assign the value 5 to pa->num

Or using a pointer to an object derived from A:

class D : public A {};

A\* pd = new D;

pd->\*pmi = 5; // assign the value 5 to pd->num

### **Declaring Pointers to Member Functions**

Pointers to data members, which are used less often than pointers to member functions.

A pointer to a member function consists of the member function's return type, the class name followed by ::, the pointer's name, and the function's parameter list.

class A

{

public:

int func ();

};

int (A::\*pmf) (); /\* pmf is a pointer to some member

function of class A that returns int and takes no

arguments\*/

pmf = &A::func; //assign pmf

A a;

A \*pa = &a;

(a.\*pmf)(); // invoke a.func()

// call through a pointer to an object

(pa->\*pmf)(); // calls pa->func()