**Pointers to class members**

Just like pointers to normal variables and functions, we can have pointers to class member functions and member variables.

**Defining a pointer of class type**

We can define pointer of class type, which can be used to point to class objects.

class Simple

{

public:

int a;

};

int main()

{

Simple obj;

Simple\* ptr; // Pointer of class type

ptr = &obj;

cout << obj.a;

cout << ptr->a; // Accessing member with pointer

}

Here you can see that we have declared a pointer of class type which points to class's object. We can access data members and member functions using pointer name with arrow -> symbol.

**Pointer to Data Members of class**

We can use pointer to point to class's data members (Member variables).

**Syntax for Declaration :**

datatype class\_name :: \*pointer\_name ;

**Syntax for Assignment :**

pointer\_name = &class\_name :: datamember\_name ;

Both declaration and assignment can be done in a single statement too.

datatype class\_name::\*pointer\_name = &class\_name::datamember\_name ;

**Using with Objects**

For accessing normal data members we use the dot . operator with object and -> qith pointer to object. But when we have a pointer to data member, we have to dereference that pointer to get what its pointing to, hence it becomes,

Object.\*pointerToMember

and with pointer to object, it can be accessed by writing,

ObjectPointer->\*pointerToMember

Lets take an example, to understand the complete concept.

class Data

{

public:

int a;

void print() { cout << "a is "<< a; }

};

int main()

{

Data d, \*dp;

dp = &d; // pointer to object

int Data::\*ptr=&Data::a; // pointer to data member 'a'

d.\*ptr=10;

d.print();

dp->\*ptr=20;

dp->print();

}

Output :

a is 10 a is 20

The syntax is very tough, hence they are only used under special circumstances.

**Pointer to Member Functions**

Pointers can be used to point to class's Member functions.

**Syntax :**

return\_type (class\_name::\*ptr\_name) (argument\_type) = &class\_name::function\_name ;

Below is an example to show how we use ppointer to member functions.

class Data

{ public:

int f (float) { return 1; }

};

int (Data::\*fp1) (float) = &Data::f; // Declaration and assignment

int (Data::\*fp2) (float); // Only Declaration

int main(0

{

fp2 = &Data::f; // Assignment inside main()

}