

NIOS Computer: Chapter 13 – Classes & Objects with Constructors/Destructors Part 3

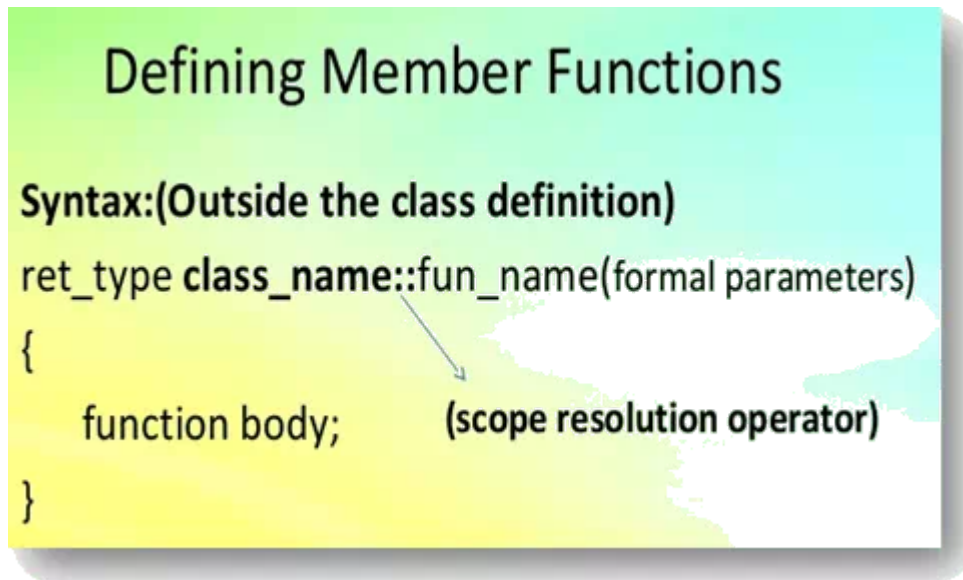


Image of Defining Member Function

2. Accessing Class Member

Through object, data member and member function present in public can be accessed. The general format is:

Object name . data member ;

Object name . member function ;

The dot operator is called the class member access operator.

3. Member Function Definition

Member function can be defined in two ways:

(i) Inside the class

(ii) Outside the class

Inside the class : When a member function is defined inside a class, it is considered to be inline by default. If a member function is very small then it should be defined inside the class.

The class declaration of previous program will be as follow:

1. **class** student
2. {
3. **char** name [20] ;

Visit examrace.com for free study material, doorsteptutor.com for questions with detailed explanations, and "Examrace" YouTube channel for free videos lectures

```
4.     int rn ;
5.     float marks ;
6.     public :
7.     void getdata ( )
8.     {
9.         cin >> name >> m >> marks ;
10.    }
11.    void putdata ( )
12.    {
13.        cout << name << rn << marks :
14.    }
15.};
```

4. Nesting of Member Functions

A member function can be called by using its name inside another member function of the same class. This is known as nesting of member functions.

The following program illustrates this concept.

```
1. class greatest
2. {
3.     int x, y, z ;
4.     public :
5.     void getdata ( ) ;
6.     void display ( ) ;
7.     int largest ( ) ;
8. };
9. int greatest : : largest ( )
10. {
11.     int T ;
```

Visit examrace.com for free study material, doorsteptutor.com for questions with detailed explanations, and "Examrace" YouTube channel for free videos lectures

```
12.     if ( x > y )
13.         T = x ;
14.     else
15.         T = y ;
16.     if ( T > z )
17.         return ( T ) ;
18.     else
19.         return ( z ) ;
20. }

21. void greatest : : getdata ( )
22. {
23.     cout << "Enter values of x, y, z" << "\n";
24.     cin >> x >> y >> z ;
25. }

26. void greatest : : display ( )
27. {
28.     cout << "largest value" << largest ( ) << "\n";
29. }

30. void main ( )
31. {
32.     greatest A;
33.     A. getdata ( ) ;
34.     A. display ( ) ;
35. }
```