13.2.2: Classes and Objects - Member Functions

Member Functions in Classes

There are 2 ways to define a member function:

- Inside class definition
- Outside class definition

To define a member function outside the class definition we have to use the scope resolution :: operator along with class name and function name.

```cpp
// C++ program to demonstrate function
// declaration outside class

#include <bits/stdc++.h>
using namespace std;

class Courses
{
    public:
        string coursename;
        int id;

        // printname is not defined inside class definition
        void printname();

        // printid is defined inside class definition
        void printid()
        {
```

https://eng.libretexts.org/Courses/Delta_College/C___Programming_I_(McClanahan)/13%3A_Object_Oriented_C/13.02%3A__…
Updated: Mon, 06 Jul 2020 11:03:07 GMT
Powered by
cout << "Course id is: " << id;
}
);

// Definition of printname using scope resolution operator ::
void Courses::printname()
{
    cout << "Course name is: " << coursename;
}

int main()
{
    Courses thisObj;
    thisObj.coursename = "CSP 31A";
    thisObj.id=976;

    // call printname()
    // This is defined outside of the class Courses
    thisObj.printname();
    cout << endl;

    // call printid()
    // This is defined inside of the class Courses
    thisObj.printid();
    return 0;
}

In this code we have 2 separate ways to provide output. One is the method printid(), which is a method of the Courses class, it is defined inside of the class definition because it is between the opening and closing brackets of the block of code which is the class definition.

We also have the method printname(), which is defined outside of the class definition. It is still a method of the class Courses, because as we can see it is specifically defined as void Courses::printname() using the scope resolution operator ::. Also, notice that we do declare printname() inside the class, but provide no body to the code.

Both of these methods belong to the class Courses, and you have to call them with the object dot manner in your code.

    // printname() method prints this statement
    Geekname is: CSP 31A

    // printid() method prints this statement
    Geek id is: 976

Note that all the member functions defined inside the class definition are by default inline, but you can also make any non-class function inline by using keyword inline with them. Inline functions are actual functions, which are copied everywhere during compilation, like pre-processor macro, so the overhead of function calling is reduced.

Note: Declaring a friend function is a way to give private access to a non-member function.

Adapted from: "C++ Classes and Objects" by Charalampos Sarantoglou, opensource.com

https://eng.libretexts.org/Courses/Delta_College/C___Programming_I_(McClanahan)/13%3A_Object_Oriented_C/13.02%3A__...