

# C++ FRIEND FUNCTIONS

[http://www.tutorialspoint.com/cplusplus/cpp\\_friend\\_functions.htm](http://www.tutorialspoint.com/cplusplus/cpp_friend_functions.htm)

Copyright © tutorialspoint.com

A friend function of a class is defined outside that class' scope but it has the right to access all private and protected members of the class. Even though the prototypes for friend functions appear in the class definition, friends are not member functions.

A friend can be a function, function template, or member function, or a class or class template, in which case the entire class and all of its members are friends.

To declare a function as a friend of a class, precede the function prototype in the class definition with keyword **friend** as follows:

```
class Box
{
    double width;
public:
    double length;
    friend void printWidth( Box box );
    void setWidth( double wid );
};
```

To declare all member functions of class ClassTwo as friends of class ClassOne, place a following declaration in the definition of class ClassOne:

```
friend class ClassTwo;
```

Consider the following program:

```
#include <iostream>

using namespace std;

class Box
{
    double width;
public:
    friend void printWidth( Box box );
    void setWidth( double wid );
};

// Member function definition
void Box::setWidth( double wid )
{
    width = wid;
}

// Note: printWidth() is not a member function of any class.
void printWidth( Box box )
{
    /* Because printWidth() is a friend of Box, it can
       directly access any member of this class */
    cout << "Width of box : " << box.width << endl;
}

// Main function for the program
int main( )
{
    Box box;

    // set box width without member function
    box.setWidth(10.0);

    // Use friend function to print the width.
```

```
    printWidth( box );  
    return 0;  
}
```

When the above code is compiled and executed, it produces the following result:

```
Width of box : 10
```