

Friend Functions in C++ | C++ Tutorials for Beginners #26

In this tutorial, we will discuss friend function in C++

Friend Function in C++

Friend functions are those functions that have the right to access the private data members of class even though they are not defined inside the class. It is necessary to write the prototype of the friend function. One main thing to note here is that if we have written the prototype for the friend function in the class it will not make that function a member of the class. An example program to demonstrate the concept of friend function is shown below.

```
class Complex{
    int a, b;
    friend Complex sumComplex(Complex o1, Complex o2);
public:
    void setNumber(int n1, int n2){
        a = n1;
        b = n2;
    }

    // Below line means that non member - sumComplex function is allowed to do anything with my private parts
    void printNumber(){
        cout<<"Your number is "<<a<<" + "<<b<<"i"<<endl;
    }
};

Complex sumComplex(Complex o1, Complex o2){
    Complex o3;
    o3.setNumber((o1.a + o2.a), (o1.b+o2.b))
    ;
    return o3;
}
```

Code Snippet 1: Complex Class

As shown in Code Snippet 1, we created a complex class that has integer “a” variable and “b” integer variable as private class members; and “setNumber” void function, “printNumber” void function as public class members. The “sumComplex” friend function prototype is written as well in the complex class. These functions are explained below.

We have defined a “setNumber” function. In this function the values are assigned to the variables “a” and “b” because they are private data members of the class and values cannot be assigned directly. We have defined a “printNumber” function. In this function, the values of the variable “a” and “b” are being printed. We have defined a “sumComplex” friend function. In this function, the object “o3” is created which calls the “setNumber” function and passes the values of two objects after performing addition on them.

```
int main(){
    Complex c1, c2, sum;
    c1.setNumber(1, 4);
    c1.printNumber();

    c2.setNumber(5, 8);
    c2.printNumber();

    sum = sumComplex(c1, c2);
    sum.printNumber();

    return 0;
}
```

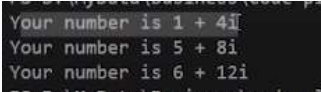
Code Snippet 2: main Program

As shown in Code Snippet 2:

- We have created object “c1”, “c2”, and “sum” of complex data-type.
- The object “c1” calls the “setNumber” and “printNumber” functions.
- The object “c2” calls the “setNumber” and “printNumber” functions.

- The function "sumComplex" is called and the values are assigned to the "sum".
- The object "sum" calls the "printNumber" functions.

The output of the following program is shown in figure 1.



```

Your number is 1 + 4i
Your number is 5 + 8i
Your number is 6 + 12i

```

Figure 1: Complex Program Output

As shown in figure 1, the output of the complex number program is printed.

Properties of Friend Function

- Not in the scope of the class
- Since it is not in the scope of the class, it cannot be called from the object of that class, for example, **sumComplex()** is invalid
- A friend function can be invoked without the help of any object
- Usually contain objects as arguments
- Can be declared under the public or private access modifier, it will not make any difference
- It cannot access the members directly by their names, it needs (object_name.member_name) to access any member.

Code as described/written in the video

```

#include<iostream>
using namespace std;

// 1 + 4i
// 5 + 8i
// -----
// 6 + 12i
class Complex{
    int a, b;
    friend Complex sumComplex(Complex o1, Complex o2);
public:
    void setNumber(int n1, int n2){
        a = n1;
        b = n2;
    }

    // Below line means that non member - sumComplex function is allowed to do anything with my private part
    void printNumber(){
        cout<<"Your number is "<<a<<" + "<<b<<"i"<<endl;
    }
};

Complex sumComplex(Complex o1, Complex o2){
    Complex o3;
    o3.setNumber((o1.a + o2.a), (o1.b + o2.b));
}

```

← Previous

Next →

