



More on C++ Friend Functions (Examples & Explanation) | C++ Tutorials for Beginners #28



[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

More on C++ Friend Functions (Examples & Explanation) | C++ Tutorials for Beginners #28

In this tutorial, we will discuss more on friend functions in C++ with examples

Friend Functions in C++

As we have already discussed in previous lectures friend functions are those functions that can access the private data members of the other class. An example program to demonstrate friend functions in C++ is shown below.

Friend Function Example 1

```
class Y;
```

```
class X{
    int data;
public:
    void setValue(int value){
        data = value;
    }
    friend void add(X, Y);
};

class Y{
    int num;
public:
    void setValue(int value){
        num = value;
    }
    friend void add(X, Y);
};

void add(X o1, Y o2){
    cout<<"Summing data of X and Y objects gives me "<< o1.data + o2.num;
}
```

Code Snippet 1: Friend Function Example 1

As shown in a code snippet 1,

- 1st class “Y” is declared at the top which is known as forward declaration to let the compiler know that this class is defined somewhere in the program.
- 2nd class “X” is defined which consists of private data member “data” and public member function “setValue” which assigns the value to the private data member “data”. At the end friend function “add” is declared.
- 3rd class “Y” is defined which consists of private data member “num” and public member function “setValue” which assigns the value to the private data member “num”. At the end friend function “add” is declared.
- 4th function “add” is defined which add the value of the objects of class “X” and “Y” and print it.

The main program is shown in Code Snippet 2.

```
int main(){
    X a1;
    a1.setValue(3);

    Y b1;
    b1.setValue(15);

    add(a1, b1);
    return 0;
}
```

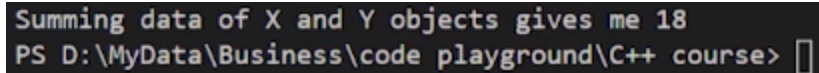
Code Snippet 2: Main Program

As shown in Code Snippet 2,

- 1st object “a1” of the data type “X” is declared
- 2nd function “setValue” is called by the object “a1” and the value “3” is passed
- 3rd object “b1” of the data type “Y” is declared

- 4th function “setValue” is called by the object “b1” and the value “15” is passed
- 5th function “add” is called and the objects “a1” and “b1” are passed to it. The function “add” will add the values of both objects and print them.

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

A terminal window with a dark background. The first line shows the output of a program: "Summing data of X and Y objects gives me 18". The second line shows the command prompt "PS D:\MyData\Business\code playground\C++ course>" followed by a cursor.

```
Summing data of X and Y objects gives me 18
PS D:\MyData\Business\code playground\C++ course> 
```

Figure 1: Program Output 1

As shown in figure 1, the sum of both values is shown which is “18”.

Friend Function Example 2

```
class c2;

class c1{
    int val1;
    friend void exchange(c1 & , c2 &);
public:
    void indata(int a){
        val1 = a;
    }

    void display(void){
        cout<< val1 <<endl;
    }
};

class c2{
```

```
int val2;
friend void exchange(c1 &, c2 &);
public:
    void indata(int a){
        val2 = a;
    }

    void display(void){
        cout<< val2 <<endl;
    }
};

void exchange(c1 &x, c2 &y){
    int tmp = x.val1;
    x.val1 = y.val2;
    y.val2 = tmp;
}
```

Code Snippet 3: Friend Function Example 2

As shown in a code snippet 3,

- 1st class “c2” is declared at the top which is known as forward declaration to let the compiler know that this class is defined somewhere in the program.
- 2nd class “c1” is defined which consists of private data member “val1” and friend function “exchange” which takes reference variables “c1” and “c2” as parameters. The public member function “indata” is defined which assigns the value to the private data member “val1” and the function “display” prints the value of the data

member “val1”.

- 3rd class “c2” is defined which consists of private data member “val2” and friend function “exchange” which takes reference variables “c1” and “c2” as parameters. The public member function “indata” is defined which assigns the value to the private data member “val2” and the function “display” prints the value of the data member “val2”.
- 4th function “exchange” is defined which swap the values.

The main program is shown in Code Snippet 4.

```
int main(){
    c1 oc1;
    c2 oc2;

    oc1.indata(34);
    oc2.indata(67);
    exchange(oc1, oc2);

    cout<<"The value of c1 after exchanging becomes: ";
    oc1.display();
    cout<<"The value of c2 after exchanging becomes: ";
    oc2.display();

    return 0;
}
```

Code Snippet 4: Main program

As shown in Code Snippet 4,

- 1st object “oc1” of the data type “c1” is declared
- 2nd object “oc2” of the data type “c2” is declared
- 3rd function “indata” is called by the object “oc1” and the value “34” is passed
- 4th function “indata” is called by the object “oc2” and the value “67” is passed
- 5th function “exchange” is called and the objects “oc1” and “oc2” are passed to it. The function “exchange” will swap both values and
- 6th function “display” is called by the objects “oc1” and “oc2” which will print their values.

The output of the following program is shown in figure 2.

```
The value of c1 after exchanging becomes: 67
The value of c2 after exchanging becomes: 34
```

Figure 2: Program Output 2

As shown in figure 2, the values are swapped.

[Previous](#)[Next](#)

CodeWithHarry

Copyright © 2022 CodeWithHarry.com

