Function Objects (Functors) In C++ STL | C++ Tutorials for Beginners #74

In the last tutorial we completed learning about some of the most commonly used containers, vector, list, map and their methods. Today we'll start with function objects in C++ STL.

What is a function object?

A function object is a function wrapped in a class so that it is available as an object.

That is, we can then use a function as an object. The question that might have been raised in your mind would be, **why to substitute a function with an object**? The answer is to make them all usable in an Object-Oriented Programming paradigm. Now what does that mean? We'll try decoding the purpose of using functions as an object via a program. So, hold onto your editors.

Understanding code snippet 1:

- Be sure to include the header file < functional> before you do anything else.
- And let's create an array of some 6 elements.
- Suppose we want to sort this array in ascending order. So we'll include a header file <algorithm> and write the syntax of the sort object which is,

```
sort(address of first element, address of last element);
```

Code Snippet 1: Syntax for sort algorithm

- And let's just sort from the beginning to the 5th element.
- And run a loop to see the resultant array.

```
#include<iostream>
#include<functional>
#include<algorithm>

using namespace std;

int main(){

    // Function Objects (Functor) : A function wrapped in a class so that it is available li
    int arr[] = {1, 73, 4, 2, 54, 7};
    sort(arr,arr+5);
    for (int i = 0; i < 6; i++)
    {
```

```
        cout<<arr[i]<<endl;
    }

    return 0;
}
```

Code Snippet 2: Program to sort an array in ascending order

Output of the above program is given below. And you'll notice that the last element remained untouched.

```
1
2
4
54
73
7
PS D:\MyData\Business\code playground\C++ course>
```

Figure 1: Output of the above program

But what if we wanted to sort the same array in descending order, since the sort function can default sort in ascending order only? So, here comes our saviour, **functional objects**. Our sort function also takes a third parameter which is a functor (functional object).

Let's see how they work via the snippet below:

- Among all the different functors we have, the one to help this sort function to sort the array in descending order, is the **greater<int>()**.

```
sort( arr, arr+6, greater< int >( ));
```

Code Snippet 1: Syntax for using a functor in an algorithm

- And that's it. Our array will now get sorted in descending order.

See the output after the above changes we made:

```
73
54
7
4
2
1
PS D:\MyData\Business\code playground\C++ course>
```

Figure 1: Output of the above program after using a functor greater<int>()

It would be unnecessarily lengthy to review all the functors, as my role was to introduce them to you and show you how they are used.

In addition, I invite you all to explore the other function objects on the site Function objects. You should go through them lightly because a lot of them would be overwhelming to you as a beginner. We use most of these functors for our STL algorithms, so you might want to check all those algorithms on Functions in <algorithm> - C++ Reference. You can go through them at your own pace. Incorporate them into your programs. Take advantage of them and use them how you see fit.

Thank you for being with me throughout the tutorial. I hope you enjoyed it. If you haven't checked out the whole playlist yet, move on to codewithharry.com or my YouTube channel to access it. You will surely enjoy them all.

Looking forward to seeing you all next time. Till then keep coding.

Previous

Next

CodeWithHarry