Tutorial 74 - Function Objects (Functors) in C++ STL

What is a Function Object (Functor)?

- A **function object (functor)** is a **function wrapped inside a class**, allowing it to be used as an **object**.
- **Purpose:**
  - Enables functions to be used within **Object-Oriented Programming (OOP)**.
  - Allows **custom behavior** when used in STL algorithms.

Sorting an Array in C++ Using `sort()`

Syntax of `sort()` Function

```
sort(start_address, end_address);

```

- By default, it sorts in **ascending order**.
- Requires `<algorithm>` header file.

Example: Sorting an Array in Ascending Order

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

using namespace std;

int main() {
    int arr[] = {1, 73, 4, 2, 54, 7};
    sort(arr, arr + 5); // Sorts first 5 elements in ascending order

    for (int i = 0; i < 6; i++) {
        cout << arr[i] << endl;
    }

    return 0;
}

```

Output:

```
1
2
4
54
73
7  // Last element remains unchanged

```

Using Function Objects (Functors) for Sorting

Why Use Functors?

- The `sort()` function **defaults to ascending order**.

- To sort in **descending order**, we pass a **functor** as the third argument.

Syntax for Using a Functor in `sort()`

```
sort(start_address, end_address, greater<int>());

```

- `greater<int>()` is a **predefined functor** in `<functional>` that sorts in **descending order**.

Example: Sorting an Array in Descending Order

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

using namespace std;

int main() {
    int arr[] = {1, 73, 4, 2, 54, 7};
    sort(arr, arr + 6, greater<int>()); // Sorts in descending order

    for (int i = 0; i < 6; i++) {
        cout << arr[i] << endl;
    }

    return 0;
}

```

Output:

```
73
54
7
4
2
1

```

📌 **Key Takeaway:** Functors allow us to **customize the behavior** of STL algorithms like `sort()`.

Additional Functors in C++ STL

Functor	Description
`greater<T>()`	Sorts in **descending order**
`less<T>()`	Sorts in **ascending order** (default behavior)
`multiplies<T>()`	Multiplies elements
`plus<T>()`	Adds elements
`minus<T>()`	Subtracts elements
`divides<T>()`	Divides elements

Short Notes on Function Objects (Functors) in C++ STL

- **Functors are objects that behave like functions**.
- Used in STL algorithms to **customize behavior**.
- **Default sorting is ascending**, but we can pass a functor to change it.
- **Example:** `greater<int>()` sorts an array in **descending order**.
- **Other functors** include `plus<T>()`, `minus<T>()`, `multiplies<T>()`, etc.
- **Explore more STL functors** from: C++ Reference - Function Objects.

Conclusion

- **Functors enhance the flexibility** of STL algorithms.
- **Practice using different function objects** in sorting, searching, and arithmetic operations.
- Keep experimenting with functors to **master their usage in STL**. 🚀