

comparator Function STL

01-03-2023

Date.....

* custom comparator:- By using comparator function, we can define custom behaviour in sorting.

For example:- If we write a string & sort it using STL sort function. Then it will sort the string in ascending order.

```
int main()
```

```
{
```

```
    string s = "vanshika";
```

```
    sort(s.begin(), s.end());
```

```
    cout << s;
```

```
    return 0;
```

```
}
```

↘ output: aahiknsv

If we want to print this string in descending order, then if we use sorting function like this:-

```
sort(s.end(), s.begin());
```

↘ by using this, nothing will print.

In this situation, we use our custom comparator function.

Basically, "comparator" function returns a value, convertible to bool, which basically tells us whether the passed "first" argument should be placed before the passed "second" argument or not.

* what is the use of comparator in sort function?

By applying the comparator function along with a sorting algorithm.

sort() is able to determine the order that the elements should be in for the sequence to be sorted.

That's all the sort() function does, it applies the comparator function to pairs of elements and rearranges them until the sequence is sorted.

Comparator function returns bool value & take 2 characters at a time,

```
bool cmp(char a, char b)
{
    return a < b;
}
```

↙ this means if a is less than b then return true.

code:-

```
bool cmp(char a, char b)
```

```
{
    return a < b;
}
```

```
int main()
```

```
{
    string s = "vanshika";
    sort(s.begin(), s.end(), cmp);
    cout << s << endl;
    return 0;
}
```



this function
will sort the
string in ascending
order.



Here we
pass our
cmp function

Now, if we want the string in descen-
ding order, then we don't need to
do any changes in sort function, just
change the < to > in comparator
function,

```
bool cmp(char a, char b)
```

```
{
    return a > b;
}
```


There is a built in comparator in code STL also,

```
sort(s.begin(), s.end(), greater<int>());
```

this is ↓

* In most cases we use custom sort.

the built in comparator to sort the string in descending order.