CodeWithHarry

HTML CSS JS C C++ JAVA PYTHON PHP

Overview Q&A Downloads Announcements

The C++ Standard Template Library (STL) | C++ Tutorials for Beginners #69

We have been waiting so long to start this, but creating a base is as important as any other phase. So, today we'll be starting the most awaited topic, the STL(Standard Template Library).
There is a reason why I've been saying that this topic is a must for all the competitive programmers out there,so let's deal with that first.

Why is this important for competitive programmers?

1. Competitive programming is a part of various environments, be it job interviews, coding contests and all, and if you're in one of those environments, you'll be given limited time to code your program.
2. So, suppose you want in your program, a resizable array, or sort an array or any other data structure. or search

for some element in your container.

3. You will always try to code a function which will execute the above mentioned things, and end up losing a great amount of time. But here is when you will use STL.

An STL is a library of generic functions and classes which saves you time and energy which you would have spent constructing for your use. This helps you reuse these well tested classes and functions umpteenth number of times according to your own convenience.

To put this simply, STL is used because it is not a good idea to reinvent something which is already built and can be used to innovate things further. Suppose you go to a company who builds cars, they will not ask you to start from scratch, but to start from where it is left. This is the basic idea behind using STL.

COMPONENTS OF STL:

We have three components in STL:

1. Containers
2. Algorithm
3. Iterators

Let's deal with them individually;

Containers:

Container is an object which stores data. We have different containers having their own benefits. These are the implemented template classes for our use, which can be used just by including this library. You can even customise these template classes.

Algorithms:

Algorithms are a set of instructions which manipulates the input data to arrive at some desired result. In STL, we have already written algorithms, for example, to sort some data structure, or search some element in an array. These algorithms use template functions.

Iterators:

Iterators are objects which refer to an element in a container. And we handle them very much similarly to a pointer. Their basic job is to connect algorithms to the container and play a vital role in manipulation of the data.

I'll give you a quick illustration of how they work combinedly.

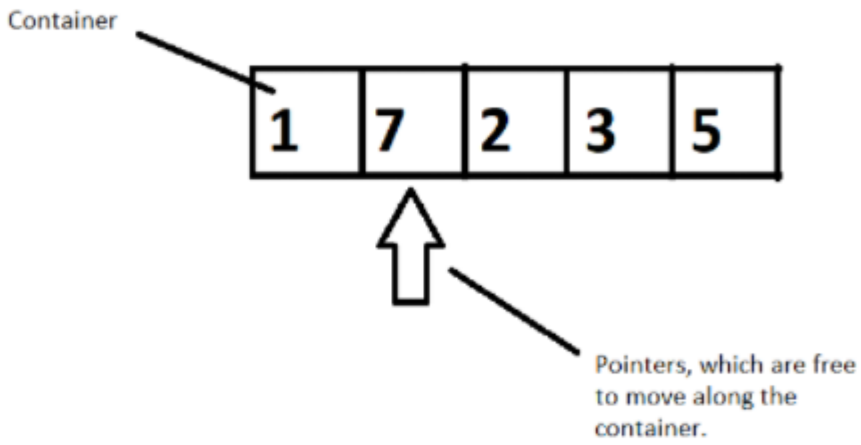

Figure 1: Illustration of how these three components work together

Suppose we have a container of integers, and we want to sort them in ascending order. We will have pointers which will help moving elements to places by pointing to it, following a well-constructed algorithm. So, here a container gets sorted by following an algorithm by the use of pointers. This is how they work in accordance with each other.

So, this was the basics of STL and the motivation behind using it in your programs. I hope I was able to introduce it to you.

Thank you, for being with me throughout, hope you liked the tutorial. If you haven't checked out the whole playlist yet, move on to [codewithharry.com](https://www.codewithharry.com) or my YouTube channel to access it. I hope you enjoy them all. See you all in the next tutorial where we'll dive deep in the containers and its different types. Till then keep coding.

Previous

Next

CodeWithHarry