

# Roadmap for C++

(From scratch to to intermediate in just 5 steps)

## ❑ STEP 1

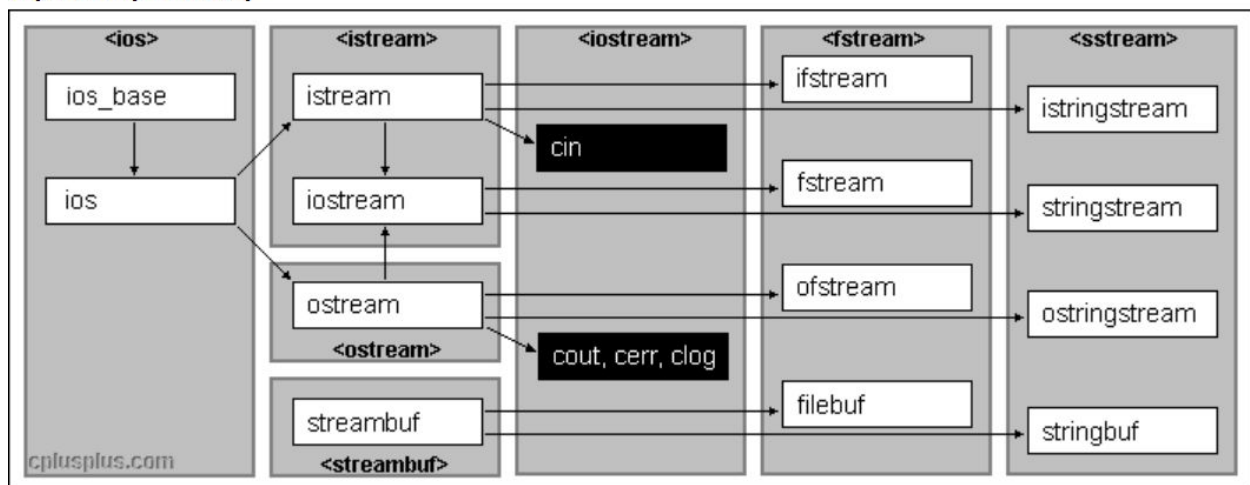
While learning a language we should be familiar with some basic syntax of input ( `cin>>` ) and output ( `cout<<` ).

After that we should know what are the variables and expressions

<http://www.cplusplus.com/reference/iolibrary/>

## Input/Output

Input/Output library



## ❑ STEP 2

One can learn basic syntax and terminology of c++ from hackerrank ( <https://www.hackerrank.com/domains/cpp> ) or hackerearth ( <https://www.hackerearth.com/practice/basic-programming/input-output/basics-of-input-output/tutorial/> ) or W3schools ( <https://www.w3schools.com/cpp/> )

And one of the evergreen resource which everyone should follow is cplusplus.com ( <http://www.cplusplus.com/doc/tutorial/> )

# C++ Language

These tutorials explain the C++ language from its basics up to the newest features introduced by C++11. Chapters have a practical orientation, with example programs in all sections to start practicing what is being explained right away.

## Introduction

- Compilers

## Basics of C++

- Structure of a program
- Variables and types
- Constants
- Operators
- Basic Input/Output

## Program structure

- Control Structures
- Functions
- Overloads and templates
- Name visibility

## Compound data types

- Arrays
- Character sequences
- Pointers
- Dynamic Memory
- Data structures
- Other data types

## Classes

- Classes (I)

## ❑ STEP 3

Now after clearing the basic concepts of C++. Now moving forward we should focus on different types of libraries C++ have.

Here is the list of that libraries with examples ( <http://www.cplusplus.com/reference/library/> )

## ● Headers

C90 (C++98)	C99 (C++11)	?
<cassert> (assert.h)	C Diagnostics Library (header)	
<cctype> (ctype.h)	Character handling functions (header)	
<cerrno> (errno.h)	C Errors (header)	
<cfloat> (float.h)	Characteristics of floating-point types (header)	
<ciso646> (iso646.h)	ISO 646 Alternative operator spellings (header)	
<climits> (limits.h)	Sizes of integral types (header)	
<locale> (locale.h)	C localization library (header)	
<cmath> (math.h)	C numerics library (header)	
<setjmp> (setjmp.h)	Non local jumps (header)	
<csignal> (signal.h)	C library to handle signals (header)	
<stdarg> (stdarg.h)	Variable arguments handling (header)	
<stdbool> (stdbool.h)	Boolean type (header)	
<stddef> (stddef.h)	C Standard definitions (header)	
<stdint> (stdint.h)	Integer types (header)	
<stdio> (stdio.h)	C library to perform Input/Output operations (header)	
<stdlib> (stdlib.h)	C Standard General Utilities Library (header)	
<string> (string.h)	C Strings (header)	
<time> (time.h)	C Time Library (header)	
<uchar> (uchar.h)	Unicode characters (header)	
<wchar> (wchar.h)	Wide characters (header)	
<wctype> (wctype.h)	Wide character type (header)	

## ❑ STEP 4

Now here is the beauty of language is Containers.

A container is a holder object that stores a collection of other objects (its elements). They are implemented as class templates, which allows a great flexibility in the types supported as elements.

There are two types of containers

- 1) Sequence containers
- 2) Associative containers

<http://www.cplusplus.com/reference/stl/>

## Container class templates

---

### Sequence containers:

<b>array</b> <small>C++11</small>	Array class (class template )
<b>vector</b>	Vector (class template )
<b>deque</b>	Double ended queue (class template )
<b>forward_list</b> <small>C++11</small>	Forward list (class template )
<b>list</b>	List (class template )

### Container adaptors:

<b>stack</b>	LIFO stack (class template )
<b>queue</b>	FIFO queue (class template )
<b>priority_queue</b>	Priority queue (class template )

### Associative containers:

<b>set</b>	Set (class template )
<b>multiset</b>	Multiple-key set (class template )
<b>map</b>	Map (class template )
<b>multimap</b>	Multiple-key map (class template )

### Unordered associative containers:

<b>unordered_set</b> <small>C++11</small>	Unordered Set (class template )
<b>unordered_multiset</b> <small>C++11</small>	Unordered Multiset (class template )
<b>unordered_map</b> <small>C++11</small>	Unordered Map (class template )
<b>unordered_multimap</b> <small>C++11</small>	Unordered Multimap (class template )

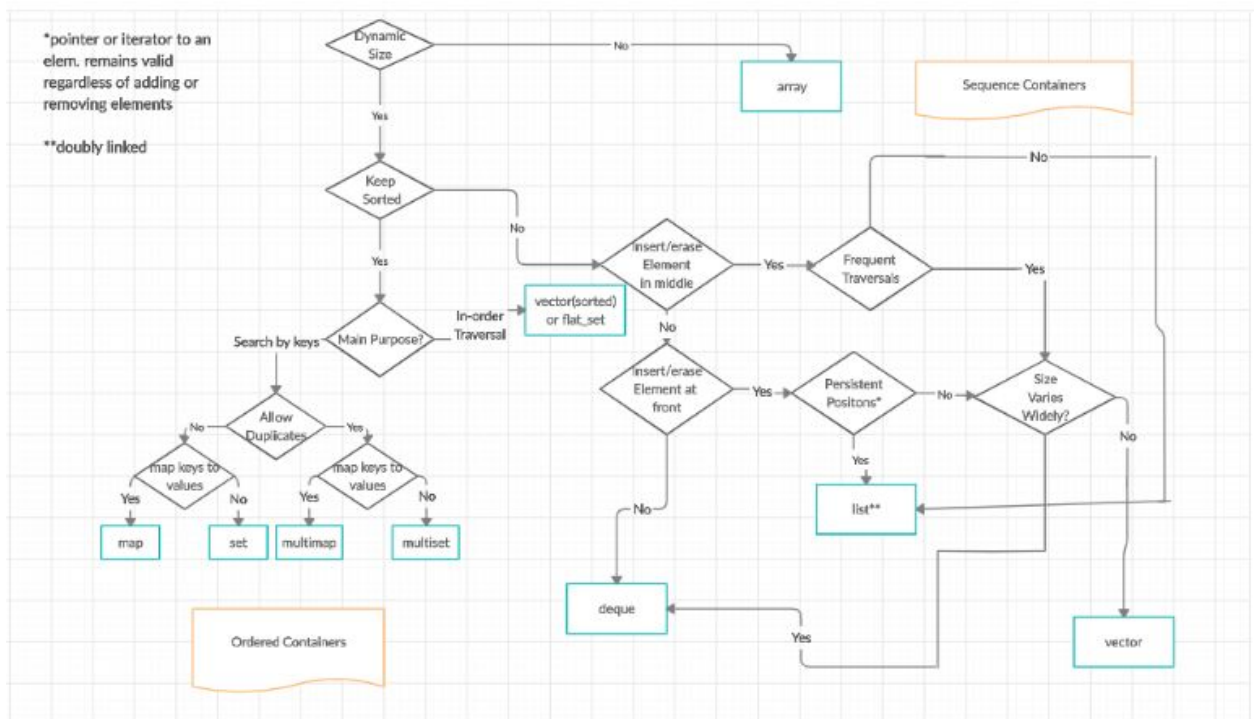
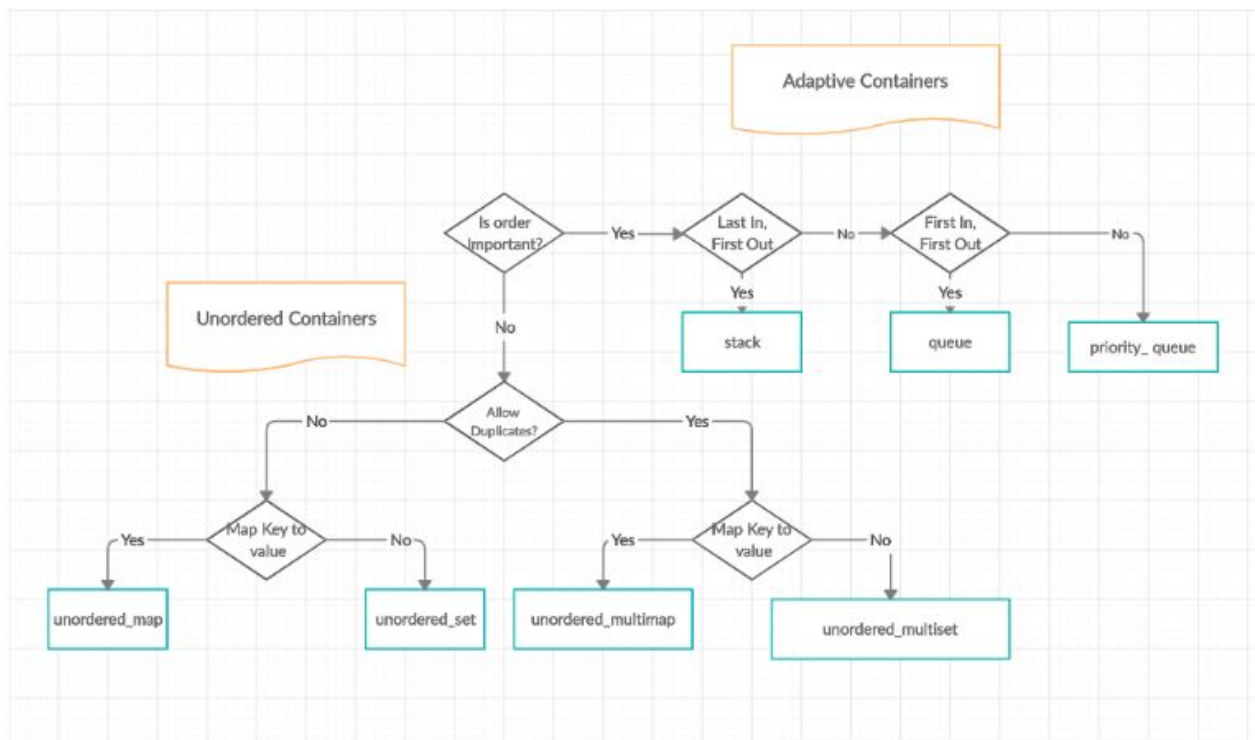
### Other:

Two class templates share certain properties with containers, and are sometimes classified with them: `bitset` and `valarray`.

## ❑ STEP 5

Learning strategy of Containers

<https://www.geeksforgeeks.org/the-c-standard-template-library-stl/>



.After learning all these things one should go for

<http://www.cplusplus.com/reference/std/>

## Handbook and Youtube Lectures

1. STL library Video lecture ( <https://www.youtube.com/watch?v=g-1Cn3ccwXY&t=1819s> )
2. CSES Problem Set ( <https://cses.fi/problemset/> )
3. The best ever handbook which is preferred by every programmer ( <https://cses.fi/book/book.pdf> )
4. CP algorithms which is well known in the world of competitive coding ( <https://cp-algorithms.com/> )

Abhishek Kumar ( 2017-21 ) : 7004149145

Nipun Khandelwal( 2018-22 ) : 9001328478

Lokesh Singh Rajawat ( 2018-22 ) : 6350485654