How to Approach Learning C++?

by Slobodan Dmitrovic

About This Talk

This talk discusses possible approaches and methodologies used in learning C++. It aims at answering the following questions:

- What is there to learn?
- In which order to learn the topics?
- How to tackle the complexity?
- How to build a solid C++ knowledge base?
- Q&A

About Slobodan Dmitrovic

- C++ software developer, trainer, and consultant
- Author of a couple of books on C and C++
- Slobodan provides C++ training courses for teams

info@cppandfriends.com

What is C++?

C++ is a programming language.

C++ is a systems programming, multi-paradigm, object-oriented, standardized programming language…

What is There To Learn?

- **The C++ language itself**
- **The C++ Standard Library**
- **Modern C++ standards**
- More…

Start with these three things, in this order.

The C++ Language

- Basic language facilities
- Classes
- Templates

} Abstraction mechanisms

The C++ Standard Library

- Containers
- Functions
- Useful Class Templates

C++ Standards

There are multiple ISO C++ standards:

- C++98
- C++03
- **C++11**
- **C++14**
- **C++17**
- C++20

Every C++ standard starting with the C++11 is informally referred to as "*Modern C++.*"

Where to Learn the C++ From?

Available resources:

- **Books**
- Online courses
- **Training with C++ trainers**
- Blogs and other online resources

C++ Knowledge Backbone

Basic Facilities

Types and Modifiers

Declaration, Definition, Initialization

Operators, Expressions, Statements

Standard Input and Output

Arrays

Pointers

References

Strings

Automatic Type Deduction

Built-in Statements

Constants

Functions

Storage, Scope, Visibility

Headers and Namespaces

Conversions

Enumerations

Lambdas and range-based loops

More…

Classes and Templates

Data Member Fields

Member Functions

Access Specifiers

Constructors

Default Constructor

Member Initialization

Copy Constructor

Copy Assignment

Move Constructor

Move Assignment

Operator Overloading

Destructors

Inheritance

Polymorphism

Introduction to Templates

More…

The C++ Standard Library

Containers

- std::vector
- std::array
- std::set
- std::map
- …

Iterators

Algorithms and Utilities

- std::sort
- std::find
- std::copy
- Min and Max
- …

More…

Modern C++ Standards

Features from C++11 to C++20

C++ is not C With Classes

Prefer resources that teach you these:

```
#include <iostream>

int main()
{
    std::cout << "Hello World!" << '\n';
}

//-----------------------------------
std::string s = "Hello World";
//-----------------------------------
class MyClass
{
};
```

To resources that teach you these:

```
#include <cstdio>

int main(void)
{
    printf("Hello World\n");
}

//-----------------------------------
const char* s = "Hello World";
char s2[] = "Hello World";
//-----------------------------------
typedef struct MyClass {
} TMyClass;
```

How Not to Approach Learning C++?

By guessing

- Do not try to learn C++ by guessing
- While some languages can be learned by playing a guessing game, C++ can not

By drawing parallels between C++ and other languages

- Try not to draw parallels between C++ and other languages, C++ is in a league of its own
- C++ is not "C with classes", nor a "subset of Java"

How to tackle the complexity?

- Break down the complexity, decide on what is important
- **We need to build a solid base first**
- We do not need to go into every detail
- **We do not need to know everything, and that is just fine**
- Avoid too much border cases and staying in dark corners
- We do not need to know the entire Standard Library by heart

Why should you learn C++?

- **It is an immensely powerful language**
- Programming in C++ can be an extremely rewarding experience
- C++ is widely used, it covers a lot of domains
- It gets you places, pays well
- A constant source of learning
- C++ developers are in **high demand**

"When I started with C++, I almost lost all interest in other languages…"

"It is an R&D engineer's paradise…"

Learn the Idiomatic C++ First

Learn the platform-agnostic, portable C++ in the beginning. Try to delegate the following topics to some other times:

- How to consume OS-specific interfaces
- The use of 3rd party libraries
- Design patterns
- Graphics
- Sounds
- Network
- (Micro)optimization

Some Challenges and Possible Solutions

- *In modern C++, the use of raw arrays and raw pointers is largely discouraged, should you learn about them?*

Learn about them if only to discourage their use in favor of std::vector, std::array and smart pointers. It is likely we will still encounter those in everyday use.

- *The std::string is not part of the basic language facilities per se, so should it be learned right away?*

The std::string is so integral to a language and everyday operation, so it is fine to learn it while learning about basic language facilities.

- *Which guidelines should you learn about in the beginning?*

Only the most important, widely used ones.

FAQ

- *What C++ version should I learn?*

It doesn't matter, as long as you have at least C++11 in mind. The language basics are almost identical in C++11, C++14, C++17 and C++20.

- *I can't learn everything there is in the language.*

Nor should you. Learn the basics, build a solid foundation first.

- *I can't learn everything there is in the C++ Standard Library.*

Nor should you. Learn only what you will be using.

- *Should I learn the C++ Standard by heart?*

Absolutely not. The C++ Standard is an instruction for compiler writers. Learn about the prominent features only.

FAQ

- *What do I need to start building C++ programs?*

A text editor and a C++ compiler.

- *What are some of the widely used C++ compilers?*

GCC, Clang, Visual C++.

- *I heard C++ was too complex.*

C++ is a tool. A tool that covers a lot of ground. It is as complex as we want it to be. C++ is nothing to be afraid of. On the contrary, it can be seen as a thing of beauty and elegance. Remember: C++ is no rocket science!

Thank you!

Climbing mountain C++ is both a challenging and rewarding task, but once at the top, the view is breathtaking.

I strongly encourage you to take on this journey of learning C++!