The C++ Master Companion — Syntax, Insight & Practice

ZephyrAmmor

October 2025

Contents

▯ Module 11: Appendices

▯ Feature Index: C++11 → C++23

A summary of major features by version.

Version	Key Features
C++11	`auto`, `nullptr`, `constexpr`, range-based for loops, move semantics, lambdas, smart pointers, uniform initialization, strongly typed enums
C++14	Generic lambdas, binary literals, relaxed constexpr, variable templates
C++17	Structured bindings, `if constexpr`, filesystem library, parallel algorithms, `std::optional`, `std::variant`, `std::any`
C++20	Concepts, ranges, coroutines, modules, `constexpr` in more contexts, `std::span`, three-way comparison (⟺)
C++23	`std::expected`, deducing `this`, improved ranges, constexpr dynamic allocation, and more library refinements

---

▯ Quick Reference Tables

Data Types

Category	Example Types	Size (Typical)
Integer	`int`, `long`, `short`, `char`	2–8 bytes
Floating Point	`float`, `double`, `long double`	4–16 bytes
Boolean	`bool`	1 byte
Character	`char`, `wchar_t`, `char16_t`, `char32_t`	1–4 bytes

Operators Summary

Category	Operators	
Arithmetic	+, -, *, /, %	
Relational	==, ≠, <, >, ⩽ , ⩾	
Logical	&&, |	|, !
Bitwise	&, |, ^, ~, <<, >>	
Assignment	=, +=, -=, *=, /=, %=	
Increment/Decrement	++, --	
Member Access	., →, ::	
Conditional	?:	

STL Containers at a Glance

Container	Type	Key Traits
`vector`	Sequence	Fast random access, dynamic resizing
`list`	Sequence	Doubly-linked list
`deque`	Sequence	Double-ended queue
`set` / `multiset`	Associative	Sorted, unique/non-unique keys
`map` / `multimap`	Associative	Key-value pairs, sorted
`unordered_map` / `unordered_set`	Hash-based	Average O(1) access

□ Compiler Flags and Build Commands

GCC/Clang

```
g++ main.cpp -std=c++20 -Wall -Wextra -O2 -o program
```

Common Flags:

- `-std=c++20` → Select language standard
- `-Wall -Wextra` → Enable warnings
- `-O2`, `-O3` → Optimization levels
- `-g` → Debug info

MSVC

```
cl /EHsc /std:c++20 main.cpp
```

Flags:

- `/EHsc` → Enable exception handling
- `/O2` → Optimize code
- `/W4` → Warning level

□ Glossary of Key Terms

Term	Meaning
RAII	Resource Acquisition Is Initialization — tie resource lifetime to object lifetime
Rvalue	Temporary object with no persistent storage
Lvalue	Object with an identifiable memory address
Undefined Behavior (UB)	Behavior not defined by the C++ standard — dangerous!
Template Instantiation	Compiler generates concrete code from a template when used
Virtual Table (vtable)	Lookup table used to resolve virtual function calls at runtime
Linker	Combines object files into a final executable

🔗 External Resources

- cppreference.com — Definitive C++ reference
- C++ ISO Standard Drafts — Official C++ specification drafts
- Compiler Explorer (godbolt.org) — Visualize compiler output
- Modern C++ Features Summary
- Books: *Effective Modern C++* (Scott Meyers), *A Tour of C++* (Bjarne Stroustrup)

🧭 How to Study This Guide

1. Layered Learning: Don't memorize syntax — understand *why* features exist.
2. Code Actively: Implement each concept immediately after learning it.
3. Debug Often: Understand error messages — they're your compiler's mentorship.
4. Visualize Memory: Especially for pointers, references, and lifetimes.
5. Refactor Constantly: Modern C++ is about elegance *and* safety.
6. Build Projects: Each module here can evolve into a mini-project.

🧠 *Remember*: *C++ mastery isn't about knowing every keyword — it's about understanding how the language thinks*.

End of Core Modules *The C++ Master Companion — Syntax, Insight & Practice* Author: ZephyrAmmor Version: 1.0 (C++11–C++23) License: MIT