

C++ How to Program: An Objects-Natural Approach, 11/e

by Paul Deitel & Harvey Deitel

PART 1

C++20 Fundamentals Quickstart & Procedural Programming

1. Intro: Test-Driving Popular, Free C++ Compilers

Intro to Hardware, Software & Internet; Test-Driving the Visual C++, GNU g++ and LLVM clang++ compilers.

2. Intro to C++20 Programming

C++ fundamentals. “Objects-Natural” (ON) approach intro—using libraries to build powerful object-oriented applications with few lines of code. ON: Manipulating `string` Objects

3. Control Statements, Part 1

Intro to C++20 text formatting. ON: Super-Sized Integers with the Boost Multiprecision Library

4. Control Statements, Part 2

ON: Precise Monetary Calculations with the Boost Multiprecision Library

5. Functions and an Intro to Function Templates

ON: Lfnylun Lhqtomh Wjtz Qarcv: Qjwazkprlm xzz Xndmwwqhlz (encrypted title for our private-key cryptography case study)

- ON = objects-natural case study.
- C++20’s “Big Four” features: Ranges, Concepts, Modules and Coroutines.
- Live-code approach: 255 complete programs with live outputs.
- Communicate with the authors at deitel@deitel.com.
- Download source code at <https://deitel.com/cpphtp11>.

PART 2

Containers, C++20 Ranges, Pointers, Strings & Files

6. arrays, vectors, Ranges and Functional-Style Programming

Intro to functional-style programming.

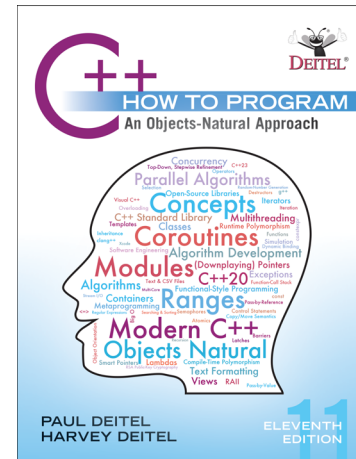
ON: Class Template `vector`

7. (Downplaying) Pointers in Modern C++

Security & safe programming. ON: C++20 spans

8. strings, string_views, Text Files, CSV Files and Regex

ON: Reading and Analyzing the Titanic Disaster Data (CSV) ON: Intro to Regular Expressions



- Static code-analysis tools.
- Use developer resources: [GitHub](https://github.com)®, [StackOverflow](https://stackoverflow.com)®, open-source, more.

PART 3

Modern Object-Oriented Programming & Exceptions

9. Custom Classes

ON: Studying the Vigenère Secret-Key Cipher Implementation

10. OOP: Inheritance and Runtime Polymorphism

Programming to an interface.

11. Operator Overloading, Copy/Move Semantics, Smart Pointers and RAII

Crafting valuable classes: Custom `MyArray` class, C++20 three-way comparison operator `<=>`, resource management via RAII (Resource Acquisition Is Initialization)

12. Exceptions and a Look Forward to Contracts

PART 4, Generic Programming: Templates, Concepts & Template Metaprogramming

13. Standard Library Containers and Iterators

Manipulating standard data structures

14. Standard Library Algorithms and C++20 Ranges & Views

Functional-style programming

15. Templates, C++20 Concepts and Metaprogramming

Compile-time polymorphism, function templates, C++20 abbreviated function templates, class templates, variadic templates, fold expressions

PART 5, Advanced Topics: Modules, Parallel Algorithms, Concurrency & Coroutines

16. C++20 Modules: Large-Scale Development

`import`, header units, module declarations, module fragments, partitions

17. Parallel Algorithms & Concurrency: A High-Level View

Multi-core performance with C++17 parallel algorithms, concurrency, multithreading

18. C++20 Coroutines

`co_yield`, `co_await`, `co_return`, coroutines support libraries, generators, executors and tasks

PART 6 Miscellaneous Topics

19. Stream I/O and C++20 Text Formatting

20. Other Topics and a Look Toward C++23 and C++26

21. Computer Science Thinking: Searching, Sorting and Big O

- Programming tips: C++ Core Guidelines, Software Engineering, Performance, Security, Errors, C++20 Modules, C++20 Concepts, Data Science.
- g++ & clang++ Docker containers.
- A look toward C++23 and C++26.
- Blog: <https://deitel.com/blog>.