C++20 for Programmers: An Objects-Natural Approach

by Paul Deitel & Harvey Deitel

PART 1
C++20 Fundamentals Quickstart & Procedural Programming

1. Intro: Test-Driving Popular, Free C++ Compilers
Object-orientation refresher. C++ on Windows®, macOS® & Linux using Visual C++®, g++, Xcode® & clang++.

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: Arbitrarily Large Integers—using free, open-source libraries

4. Control Statements, Part 2
ON: Using the `miniz-cpp` Library to Read and Write ZIP File Objects

5. Functions and an Intro to Function Templates
ON: Lnfylun Lhqtomh Wjtz Qarcv: Qjwazkrplm xzz Xndmwwqhlz (encrypted title for our private-key cryptography case study)

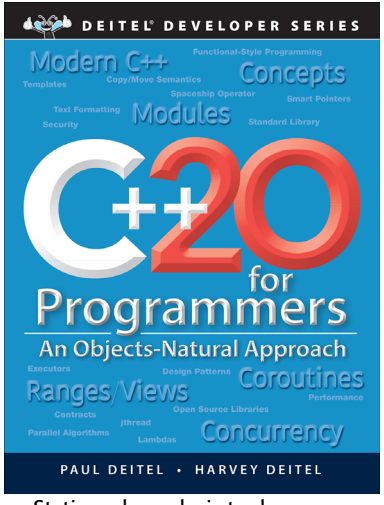
- Shows C++ as it's intended to be used for building **business-critical** and **mission-critical systems.**
- C++20's "Big Four" features: **Ranges**, **Concepts**, **Modules** and **Coroutines**.
- **Live-code approach**: Hundreds of complete programs with live outputs.
- Communicate with the authors at **`deitel@deitel.com`**.

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 `span`s

8. strings, string_views, Text Files, CSV Files and Regex
ON: Reading and Analyzing the Titanic Disaster Data (CSV)



- Static code-analysis tools.
- Download source code at **`https://deitel.com/cpp20fp`**.

PART 3
Modern Object-Oriented Programming & Exceptions

9. Custom Classes
ON: Object serialization with JSON

10. OOP: Inheritance and Runtime Polymorphism
Program to an interface, non-virtual interface idiom (NVI), `std::variant/std::visit`

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
Online Miscellaneous Topics

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

20. (Online) Other Topics and a Look Toward C++23

- Margin icons identify **Modern C++ features** by version: **C++20**, **C++17**, **C++14**, **C++11** & forthcoming **C++23**.
- Margin icons identify programming tips: **C++ Core Guidelines**, **Software Engineering**, **Performance**, **Security**, **Errors**, **C++20 Modules**.
- Use developer resources: **GitHub®**, **StackOverflow®**, **open-source**, more.
- GNU g++ & Clang **Docker containers**.
- A look toward **C++23**.
- Blog: **`https://deitel.com/blog`**.