Best Practices for Modern C++

C++

CORE GUIDELINES EXPLAINED

RAINER GRIMM

C++ Core Guidelines Explained

C++ Core Guidelines Explained: Best Practices for Modern C++

Table of Contents

- Cover
- Half Title
- Title Page
- Copyright Page
- Contents
- List of selected C++ Core Guidelines
- List of figures
- List of tables
- Foreword
- Preface
- Acknowledgments
- About the author
- Part I: The Guidelines
    - Chapter 1: Introduction
        - Target readership
        - Aim
        - Non-aims
        - Enforcement
        - Structure
        - Major sections
    - Chapter 2: Philosophy

Pearson

Table of Contents

- Chapter 3: Interfaces
  - The curse of non-const global variables
  - Dependency injection as a cure
  - Making good interfaces
  - Related rules
- Chapter 4: Functions
  - Function definitions
    - Good names
  - Parameter passing: in and out
  - Parameter passing: ownership semantics
  - Value return semantics
    - When to return a pointer (T*) or an lvalue reference (T&)
  - Other functions
    - Lambdas
  - Related rules
- Chapter 5: Classes and Class Hierarchies
  - Summary rules
  - Concrete types
  - Constructors, assignments, and destructors
    - Default operations
    - Constructor
    - Copy and move
    - Destructors
    - Other default operations
  - Class hierarchies
    - General rules
    - Designing classes
    - Accessing objects
  - Overloading and overloaded operators
    - Conventional usage
  - Unions

Pearson

Table of Contents

Related rules

Chapter 6: Enumerations

General rules

Related rules

Chapter 7: Resource Management

General rules

Allocation and deallocation

Smart pointers

Basic usage

Function parameters

Related rules

Chapter 8: Expressions and Statements

General

Declarations

Names

Variables and their initialization

Macros

Expressions

Complicated expressions

Pointers

Order of evaluation

Conversions

Statements

Iteration statements

Selection statements

Arithmetic

Arithmetic with signed/unsigned integers

Typical arithmetic errors

Related rules

Chapter 9: Performance

Wrong optimizations

Table of Contents

- Wrong assumptions
- Enable optimization
- Related rules

Chapter 10: Concurrency

- General guidelines
- Concurrency
  - Locks
  - Threads
  - Condition variables
  - Data sharing
  - Resources
  - Overlooked danger
- Parallelism
- Message passing
  - Sending a value, or an exception
  - Sending a notification
- Lock-free programming
- Related rules

Chapter 11: Error Handling

- Design
  - Communication
  - Invariants
- Implementation
  - Dos
  - Donts
- If you cant throw
- Related rules

Chapter 12: Constants and Immutability

- Use const
- Use constexpr

Chapter 13: Templates and Generic Programming

Table of Contents

Use

Interfaces

Advantages of function objects

Definition

Alternative implementations with specializations

Hierarchies

Variadic templates

Perfect forwarding

Variadic templates

Metaprogramming

Template metaprogramming

Type-traits library

Constant expressions

Other rules

Related rules

Chapter 14: C-Style Programming

Entire source code available

Entire source code not available

Chapter 15: Source Files

Interface and implementation files

Namespaces

Chapter 16: The Standard Library

Containers

Text

Input and output

Related rules

Part II: Supporting Sections

Chapter 17: Architectural Ideas

Chapter 18: Nonrules and Myths

Chapter 19: Profiles

Pearson

Table of Contents

Pro.typeType safety

Pro.boundsBounds safety

Pro.lifetimeLifetime safety

Chapter 20: Guidelines Support Library

Views

Ownership pointers

Assertions

Utilities

Part III: Appendixes

Appendix A: Enforcing the C++ Core Guidelines

Visual Studio

clang-tidy

Appendix B: Concepts

Appendix C: Contracts

Index