C++ Template Metaprogramming

Concepts, Tools, and Techniques from Boost and Beyond

David Abrahams
Aleksey Gurtovoy

C++ In-Depth Series • Bjarne Stroustrup

C++ Template Metaprogramming

C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond, Portable Documents

Table of Contents

Contents

Preface

Acknowledgments

Making the Most of This Book

Chapter 1 Introduction

- 1.1 Getting Started
- 1.2 So Whats a Metaprogram?
- 1.3 Metaprogramming in the Host Language
- 1.4 Metaprogramming in C++
- 1.5 Why Metaprogramming?
- 1.6 When Metaprogramming?
- 1.7 Why a Metaprogramming Library?

Chapter 2 Traits and Type Manipulation

- 2.1 Type Associations
- 2.2 Metafunctions
- 2.3 Numerical Metafunctions
- 2.4 Making Choices at Compile Time
- 2.5 A Brief Tour of the Boost Type Traits Library
- 2.6 Nullary Metafunctions

Pearson

Table of Contents

2.7 Metafunction Definition

2.8 History

2.9 Details

2.10 Exercises

Chapter 3 A Deeper Look at Metafunctions

3.1 Dimensional Analysis

3.2 Higher-Order Metafunctions

3.3 Handling Placeholders

3.4 More Lambda Capabilities

3.5 Lambda Details

3.6 Details

3.7 Exercises

Chapter 4 Integral Type Wrappers and Operations

4.1 Boolean Wrappers and Operations

4.2 Integer Wrappers and Operations

4.3 Exercises

Chapter 5 Sequences and Iterators

5.1 Concepts

5.2 Sequences and Algorithms

5.3 Iterators

5.4 Iterator Concepts

5.5 Sequence Concepts

5.6 Sequence Equality

5.7 Intrinsic Sequence Operations

5.8 Sequence Classes

5.9 Integral Sequence Wrappers

Table of Contents

5.10 Sequence Derivation
5.11 Writing Your Own Sequence
5.12 Details
5.13 Exercises
Chapter 6 Algorithms
6.1 Algorithms, Idioms, Reuse, and Abstraction
6.2 Algorithms in the MPL
6.3 Inserters
6.4 Fundamental Sequence Algorithms
6.5 Querying Algorithms
6.6 Sequence Building Algorithms
6.7 Writing Your Own Algorithms
6.8 Details
6.9 Exercises
Chapter 7 Views and Iterator Adaptors
7.1 A Few Examples
7.2 View Concept
7.3 Iterator Adaptors
7.4 Writing Your Own View
7.5 History
7.6 Exercises
Chapter 8 Diagnostics
8.1 Debugging the Error Novel
8.2 Using Tools for Diagnostic Analysis
8.3 Intentional Diagnostic Generation
8.4 History

Pearson

Table of Contents

8.5 Details

8.6 Exercises

Chapter 9 Crossing the Compile-Time/Runtime Boundary

9.1 for_each

9.2 Implementation Selection

9.3 Object Generators

9.4 Structure Selection

9.5 Class Composition

9.6 (Member) Function Pointers as Template Arguments

9.7 Type Erasure

9.8 The Curiously Recurring Template Pattern

9.9 Explicitly Managing the Overload Set

9.10 The sizeof Trick

9.11 Summary

9.12 Exercises

Chapter 10 Domain-Specific Embedded Languages

10.1 A Little Language . . .

10.2 . . . Goes a Long Way

10.3 DSLs, Inside Out

10.4 C++ as the Host Language

10.5 Blitz++ and Expression Templates

10.6 General-Purpose DSELs

10.7 The Boost Spirit Library

10.8 Summary

10.9 Exercises

Chapter 11 A DSEL Design Walkthrough

Table of Contents

11.1 Finite State Machines

11.2 Framework Design Goals

11.3 Framework Interface Basics

11.4 Choosing a DSL

11.5 Implementation

11.6 Analysis

11.7 Language Directions

11.8 Exercises

Appendix A: An Introduction to Preprocessor Metaprogramming

A.1 Motivation

A.2 Fundamental Abstractions of the Preprocessor

A.3 Preprocessor Library Structure

A.4 Preprocessor Library Abstractions

A.5 Exercise

Appendix B: The typename and template Keywords

B.1 The Issue

B.2 The Rules

Appendix C: Compile-Time Performance

C.1 The Computational Model

C.2 Managing Compilation Time

C.3 The Tests

Appendix D: MPL Portability Summary

Bibliography

Index

Pearson

Table of Contents

Pearson