Table of Contents

About the Author

Foreword

Table of Contents

Preface

1 The First Step

- 1.1 C++ Standard Library
- 1.2 Preprocessor Directives and Header Files
- 1.3 Statements and Comments
- 1.4 Functions
- 1.5 Programming Cycle

2 Fundamentals

- 2.1 Data Types
  - 2.1.1 Storage size
  - 2.1.2 Limits
  - 2.1.3 Conversions between Data Types
- 2.2 Literal Constants
  - 2.2.1 Literal Numbers
  - 2.2.2 Literal Characters
  - 2.2.3 String Literals
- 2.3 Variables
  - 2.3.1 Variable Names
  - 2.3.2 Declaring Variables
- 2.4 Pointers
  - 2.4.1 Special Pointers
  - 2.4.2 Pointer Arithmetic
  - 2.4.3 Pointers Summary

2.5 References

2.6 Named Constants

- 2.6.1 Using #define
- 2.6.2 const Variables
- 2.6.3 Enumerations

2.7 Operators

- 2.7.1 Assignment Operator
- 2.7.2 Arithmetic Operators
- 2.7.3 Updating Assignment Operators
- 2.7.4 Increment and Decrement Operators
- 2.7.5 Relational Operators
- 2.7.6 Logical Operators
- 2.7.7 sizeof Operator
- 2.7.8 Operator Precedence and Associativity

2.8 Expressions

2.9 Statements

2.10 Scope and Lifetime of Variables

2.11 Making Decisions

- 2.11.1 The if Statement
- 2.11.2 If Statement Example
- 2.11.3 The switch Statement
- 2.11.4 Switch Statement Example

2.12 Performing Repetitive Tasks

- 2.12.1 The for Loop
- 2.12.2 The do-while Loop
- 2.12.3 The while Loop
- 2.12.4 The break and continue Statements

2.13 Arrays

- 2.13.1 Single Dimension Array
- 2.13.2 Multidimensional Arrays

2.13.3 Pointers and Arrays
2.14 Strings
2.15 C++ Containers
2.15.1 Vector
2.15.2 The string class
2.15.3 Map
2.15.4 Iterators
2.16 Dynamic Memory Allocation
2.17 Smart Pointers
2.17.1 The unique_ptr
2.17.2 The shared_ptr
2.18 Memory Allocation Example
3 Functions
3.1 Function Declaration and Definition
3.2 Return Values
3.3 Function Parameters
3.4 Default Values for Parameters
3.5 Overloaded Functions
3.6 Standard C++ Library Functions
4 Input and Output
4.1 Introduction
4.2 Iostreams
4.3 Standard Input and Output
4.4 Stream Formatting
4.4.1 Field Width
4.4.2 Justification
4.4.3 Displaying Numeric Values
4.4.4 Displaying bool values
4.5 File Input and Output

4.5.1 Opening files
4.5.2 Closing Files
4.5.3 Moving Within Files
4.5.4 Binary File Input and output
4.6 String Input and Output
4.6.1 Input String Streams
4.6.2 Output String Streams
4.7 Redirecting Streams
4.8 Stream Buffer
4.9 Stream Error Processing
4.10 Input and Output using Iterators
4.11 File I/O Example
5 Object Oriented Programming
5.1 Introduction
5.2 Fundamentals
5.2.1 Abstraction
5.2.2 Information Hiding
5.2.3 Encapsulation
5.2.4 Friends of the Class
5.3 Introducing Classes
5.3.1 Creating Objects
5.3.2 The copy constructor
5.3.3 Destroying Objects
5.3.4 Pointers and References to Objects
5.3.5 The this Pointer
5.3.6 Arrays of Objects
5.3.7 const Objects
5.4 Implementing the Chair Class
5.4.1 Class Definition

5.4.2 Implementing Constructors and Destructor

5.4.3 Implementing Other Methods

5.5 Using the Chair Class

5.6 Adding to the Chair Class

5.6.1 Class Variables and Methods

5.6.2 A Better Chair

5.7 Hiding Implementation Details

5.7.1 Chair Class definition

5.7.2 Chair_Impl Class Definition

5.7.3 ChairProperties Enumerations

5.7.4 Chair Class Implementation

5.7.5 Chair_Impl Class Implementation

5.7.6 Using the new Chair Class

5.8 Summary

6 Reusing Code

6.1 Introduction

6.2 Inheritance

6.2.1 Specifying Inheritance

6.2.2 Types of Inheritance

6.2.3 Multiple Inheritance

6.2.4 Overriding Base Class Members

6.2.5 Derived Class Constructors

6.2.6 Derived Class Destructors

6.2.7 Example: A SmartChair Class

6.2.8 Object Slicing

6.3 Composition

6.3.1 Example: A RecliningChair Class

6.4 Composition or Inheritance?

7 Polymorphism

7.1 What is Polymorphism

7.2 Virtual Functions

7.2.1 Virtual Flowers

7.3 Abstract Base Classes

7.4 Modelling Common Behavior

7.5 Interfaces and Composition

7.5.1 The Interfaces

7.5.2 The Concrete Classes

7.5.3 Using Interfaces

7.6 Interfaces and Inheritance

7.7 Guidelines for Class Design

8 Operator Overloading

8.1 Operator Overloading and Conversions

8.2 Overloaded Operators Example

8.2.1 The Assignment Operator =

8.2.2 The Insertion Operator <<

8.2.3 The Extraction Operator >>

8.2.4 Using the Overloaded Operators

8.3 Overloading Arithmetic Operators

8.4 Class Conversions

8.4.1 Convert from Another Type to Your class

8.4.2 Convert Your class to Another Type

8.4.3 Ambiguous Conversions

9 Handling Errors

9.1 Exceptions

9.2 Exception classes

9.3 Basic Exception Handling

9.4 Rethrowing an Exception

9.5 Cleaning Up Resources

9.6 Example: Using Exceptions

9.7 Exception Specification

9.8 Program Termination

9.9 Guidelines for Using Exceptions

10 Bibliography