

Table of Contents

This is not intended as a map of things that are already written, but a path of things to work on or that I already have. It's a huge task to tackle a book and there's a lot of stuff I want to cover. I even keep coming upon things I forgot to mention in earlier chapters so first I make a change here before going back (just in case I forget). In other words, this is to keep my in line so I don't ramble off into complete gibberish or start writing about things out of there chapter context.

Part 1: Introduction to Computers and Programming

1. Machines: Input/Output, Storage, and Logic
 - a. Introduction
 - b. The Three Parts of Machines
 - c. A Simple Example
 - d. Input/Output
 - e. Storage
 - f. Logic
 - g. Hardware
2. Logic and Languages
 - a. Software
 - b. Operating Systems
 - c. Languages
 - d. Machine Language
 - e. Assembly Language
 - f. High-Level Languages
 - g. Compiled Languages
 - h. Interpreted Languages
 - i. Methods of Programming
 - i. Procedural
 - ii. Object-Oriented
3. Numbering Systems
 - a. Introduction
 - b. The Decimal Numbering System
 - c. Bases Loaded
 - d. The Imaginary Numbering System (Fivemal)
 - e. Distinguishing Numbering Systems
 - f. Converting Between Numbering Systems
 - g. The Binary Numbering System (base 2)
 - h. The Hexadecimal Numbering System (base 16)
 - i. The Dead Numbering System (base 8)
4. Storage
 - a. Computers Use Binary
 - b. N-Bit CPU
 - c. Hexadecimal Saves Us
 - d. Multiple Byte Data Storage

5. The C++ Language
 - a. Choosing a Language
 - b. History of C++
- Part 2: Beginning C++ and Programming
1. Prepare Yourself
 - a. What You Need
 - b. Text Editor
 - c. C++ Software
 - d. Command Console
 2. First Steps
 - a. Breaking into the Circle
 - b. The 'Hello World' Program
 - c. Syntax
 - d. My Dear Aunt Sally (operator precedence)
 - e. Outputting Numbers
 - f. Integers
 - g. Floating Point
 - h. Limits
 - i. White Space
 - j. Comments
 3. Variables
 - a. What is a Variable
 - b. Data Types
 - c. Declaring a Variable
 - d. Declaring Multiple Variables
 - e. Identifiers
 - f. C++ Keywords
 - g. Literals
 - h. Constants
 - i. Assignment
 - j. Basic Arithmetic
 - k. Casting
 - l. Output
 - m. Input
 - n. Increment and Decrement
 - o. Land Plot Analogy
 4. Blocks and Flow
 - a. Statement Blocks
 - b. Scope and Nesting
 - c. Locals
 - d. Globals
 - e. Namespaces
 - f. Flow Control
 - g. If
 - h. Relational and Equality
 - i. Logical Operators

- j. Complex Conditions
- k. Else
- l. Looping
- m. While
- n. Do-While
- o. For
- p. Breaking and Continuing
- 5. References and Pointers
 - a. Introduction
 - b. Reference Variables
 - c. Memory Address
 - d. Pointers
 - e. Dereference
 - f. Pointer Types
 - g. Void Pointers
 - h. Null Pointers
- 6. Structures
 - a. Introduction
 - b. Structures
 - c. Struct
 - d. Creating Struct Variables
 - e. Initialize Struct Variables
 - f. Using Struct Variables
 - g. References and Structures
 - h. Pointers and Structures
 - i. Bit-Fields
 - j. Unions
 - k. Structure Padding
- 7. Functions
 - a. Introduction
 - b. Modularization
 - c. Prototyping and Definition
 - d. First Example
 - e. Return
 - f. Void
 - g. Correct Main
 - h. Parameters
 - i. Passing by Value
 - j. Passing by Reference
 - k. Constant Parameters
 - l. Scope
 - m. Call Stack
- 8. Arrays
 - a. Introduction
 - b. Array Declaration
 - c. Looping Subscripts

- d. Initializing Arrays
 - e. Declaring Arrays without Size
 - f. Copying an Array
 - g. Arrays in Memory
 - h. Pointers to Arrays
 - i. Pointer Arithmetic
 - j. References to Arrays
 - k. Constant Arrays
 - l. Array Function Parameters
 - m. Passing Arrays to Functions
 - n. SizeOf Array
9. Strings
- a. Text is Character Strings
 - b. Character Literals
 - c. ASCII and Assumptions
 - d. Char
 - e. Character Functions
 - f. String Literals and Constants
 - g. Variable Strings
 - h. Initializing Strings
 - i. Pointers to Strings
 - j. Cin and Strings
 - k. String Assignment
 - l. SizeOf String
 - m. Concatenation
 - n. Comparing Strings
 - o. Converting Strings to Numbers
 - p. Changing String Case
 - q. Unicode Strings
10. Pre-Processing
- a. The “Other” Compiler
 - b. #include
 - c. Custom Headers
 - d. Macros
 - e. Destroying Macros
 - f. Flow Control
 - g. Existence Checking
 - h. Parenthesis Unnecessary
 - i. Raising Errors
 - j. Concise Types
11. Classes and Objects
- a. Blobs with Class
 - b. Simple Declaration and Usage
 - c. Member Functions
 - d. Constructors and Destructors
 - e. Custom Constructors

- f. Initializing Member Variables
 - g. Inheritance
 - h. Overloading
 - i. Class Namespaces
 - j. Static Members
 - k. Access Modifiers
 - l. Friends
 - m. This
 - n. Structures are Classes
12. Files and I/O
- a. Console I/O (**cout/cin**)
 - b. Escape Characters
 - c. File I/O (**ofstream/ifstream**)
 - d. Binary Files
 - e. Reading in a Structure

Part 3: Advanced C++ and Programming

- 1. Memory
 - a. Stack Memory / Call Stack
 - b. Heap Memory
 - c. Allocating and Freeing Memory
- 2. Name Spaces
 - a. Global
 - b. std
- 3. Advanced Flow Control
 - a. Switch
 - b. Else If
 - c. Comma Operator
 - d. Ternary Operator
- 4. Bit Manipulation
 - a. Bit Variables in Objects
 - b. Bitwise Operators
 - c. Shifting
 - d. Making Flags
 - e. Using Enum/Define for Flag Constants
- 5. Modularization
 - a. Header Files
 - b. Multiple Source Files
 - c. Static Code Libraries
 - d. Dynamic Code Libraries (DLL, SO)
 - e. **extern** Keyword
- 6. Recursion
 - a. Show Folder and all Sub-Folders
 - b. Towers of Hanoi
- 7. Advanced Functions
 - a. Variable Arguments
 - b. Pointers to Functions

- c. Callbacks
- d. Functions inside Functions
- e. Templates
- f. Function Overloading
- g. Operator Overloading

Part 4: Object Orientation with Classes

- 1. Introduction to Classes
 - a. Versus struct
- 2. Member Functions
 - a. Pointers to
- 3. Protected, Private, and Public
- 4. Member Scope
- 5. Operator Overloading
 - a. making my string class act like a char*

Part 5: Performance and Tuning

- 1. Data Alignment
- 2. Cached and Volatile Variables
 - a. What is Caching?
 - b. Cached Variables: **register**
 - c. Non-Cached Variables: **volatile**
- 3. Outline and Inline Functions
- 4. Name Spaces
- 5. Unicode
- 6. Etiquette
 - a. Conventions

Part 6: Debugging

- 1. Tracing
- 2. Assertions
- 3. Using a Debugger

Part 7: Standard C++ and the STL

- 1. Correct C++
 - a. Default Name Space
 - b. A Politically Correct 'Hello World'
- 2. Intro to STL
- 3. Vectors and Iterators
- 4. Strings

Part 8: Multi-Threading

- 1. Synchronization Nightmare