BJARNE STROUSTRUP

The Design and Evolution of C++
Design and Evolution of C++, The

Table of Contents

Cover
Title Page
Copyright Page
Contents
Preface
Acknowledgments
Notes to the Reader
  Introduction
  How to Read this Book
  C++ Timeline
  Focus on Use and Users
  Programming Languages
  References
Part I
  The Prehistory of C++
    1.1 Simula and Distributed Systems
    1.2 C and Systems Programming
    1.3 General Background
  C with Classes
    2.1 The Birth of C with Classes
    2.2 Feature overview
    2.3 Classes
    2.4 Run-Time Efficiency
# Table of Contents

2.5 The Linkage Model  
2.6 Static Type Checking  
2.7 Why C?  
2.8 Syntax Problems  
2.9 Derived Classes  
2.10 The Protection Model  
2.11 Run-Time Guarantees  
2.12 Minor Features  
2.13 Features Considered, but not Provided  
2.14 Work Environment  

## The Birth of C++

3.1 From C with Classes to C++  
3.2 Aims  
3.3 Cfront  
3.4 Language Features  
3.5 Virtual Functions  
3.6 Overloading  
3.7 References  
3.8 Constants  
3.9 Memory Management  
3.10 Type Checking  
3.11 Minor Features  
3.12 Relationship to Classic C  
3.13 Tools for Language Design  
3.14 The C++Programming Language (1st edition)  
3.15 The Whatis? Paper  

### C++ Language Design Rules

4.1 Rules and Principles  
4.2 General Rules  
4.3 Design Support Rules
# Table of Contents

4.4 Language-Technical Rules  
4.5 Low-Level Programming Support Rules  
4.6 A Final Word  

**Chronology 1985-1993**  
5.1 Introduction  
5.2 Release 2.0  
5.3 The Annotated Reference Manual  
5.4 ANSI and ISO Standardization  

**Standardization**  
6.1 What is a Standard?  
6.2 How does the Committee Operate?  
6.3 Clarifications  
6.4 Extensions  
6.5 Examples of Proposed Extensions  

**Interest and Use**  
7.1 The Explosion in Interest and Use  
7.2 Teaching and Learning C++  
7.3 Users and Applications  
7.4 Commercial Competition  

**Libraries**  
8.1 Introduction  
8.2 C++ Library Design  
8.3 Early Libraries  
8.4 Other Libraries  
8.5 A Standard Library  

**Looking Ahead**  
9.1 Introduction  
9.2 Retrospective  
9.3 Only a Bridge?  
9.4 What Will Make C++ Much More Effective?
Table of Contents

Part II

Memory Management
10.1 Introduction
10.2 Separating Allocation and Initialization
10.3 Array Allocation
10.4 Placement
10.5 Deallocation Problems
10.6 Memory Exhaustion
10.7 Automatic Garbage Collection

Overloading
11.1 Introduction
11.2 Overload Resolution
11.3 Type-Safe Linkage
11.4 Object Creation and Copying
11.5 Notational Convenience
11.6 Adding Operators to C++
11.7 Enumerations

Multiple Inheritance
12.1 Introduction
12.2 Ordinary Base Classes
12.3 Virtual Base Classes
12.4 The Object Layout Model
12.5 Method Combination
12.6 The Multiple Inheritance Controversy
12.7 Delegation
12.8 Renaming
12.9 Base and Member Initializers

Class Concept Refinements
13.1 Introduction
Table of Contents

13.2 Abstract Classes
13.3 const Member Functions
13.4 Static Member Functions
13.5 Nested Classes
13.6 Inherited::
13.7 Relaxation of Overriding Rules
13.8 Multi-methods
13.9 Protected Members
13.10 Improved Code Generation
13.11 Pointers to Members

Casting
14.1 Major Extensions
14.2 Run-Time Type Information
14.3 A New Cast Notation

Templates
15.1 Introduction
15.2 Templates
15.3 Class Templates
15.4 Constraints on Template Arguments
15.5 Avoiding Code Replication
15.6 Function Templates
15.7 Syntax
15.8 Composition Techniques
15.9 Template Class Relationships
15.10 Template Instantiation
15.11 Implications of Templates

Exception Handling
16.1 Introduction
16.2 Aims and Assumptions
16.3 Syntax
# Table of Contents

16.4 Grouping  
16.5 Resource Management  
16.5 Resumption vs. Termination  
16.5 Asynchronous Events  
16.6 Multi-level Propagation  
16.7 Static Checking  
16.8 Invariants

Namespaces
  17.1 Introduction  
  17.2 The Problem  
  17.3 Ideals for a Solution  
  17.4 The Solution: Namespaces  
  17.5 Implications for Classes  
  17.6 C Compatibility

The C Preprocessor
  18.1 Cpp

Index