Bjarne Stroustrup
The Creator of C++

Programming
Principles and Practice Using C++

Second Edition
Table of Contents

Cover
Title Page
Copyright Page
Acknowledgments
Contents
Preface
Chapter 0 Notes to the Reader
  0.1 The structure of this book
      0.1.1 General approach
      0.1.2 Drills, exercises, etc.
      0.1.3 What comes after this book?
  0.2 A philosophy of teaching and learning
      0.2.1 The order of topics
      0.2.2 Programming and programming language
      0.2.3 Portability
  0.3 Programming and computer science
  0.4 Creativity and problem solving
  0.5 Request for feedback
  0.6 References
  0.7 Biographies
      Bjarne Stroustrup
Table of Contents

Lawrence Pete Petersen

Chapter 1 Computers, People, and Programming

1.1 Introduction
1.2 Software
1.3 People
1.4 Computer science
1.5 Computers are everywhere
   1.5.1 Screens and no screens
   1.5.2 Shipping
   1.5.3 Telecommunications
   1.5.4 Medicine
   1.5.5 Information
   1.5.6 A vertical view
   1.5.7 So what?
1.6 Ideals for programmers

Part I: The Basics

Chapter 2 Hello, World!

   2.1 Programs
   2.2 The classic first program
   2.3 Compilation
   2.4 Linking
   2.5 Programming environments

Chapter 3 Objects, Types, and Values

   3.1 Input
   3.2 Variables
   3.3 Input and type
   3.4 Operations and operators
   3.5 Assignment and initialization
Table of Contents

3.6 Composite assignment operators
3.7 Names
3.8 Types and objects
3.9 Type safety

Chapter 4 Computation
4.1 Computation
4.2 Objectives and tools
4.3 Expressions
4.4 Statements
4.5 Functions
4.6 vector
4.7 Language features

Chapter 5 Errors
5.1 Introduction
5.2 Sources of errors
5.3 Compile-time errors
5.4 Link-time errors
5.5 Run-time errors
5.6 Exceptions
5.7 Logic errors
5.8 Estimation
5.9 Debugging
5.10 Pre- and post-conditions
5.11 Testing

Chapter 6 Writing a Program
6.1 A problem
6.2 Thinking about the problem
6.3 Back to the calculator!
6.4 Grammars
# Table of Contents

6.5 Turning a grammar into code  
6.6 Trying the first version  
6.7 Trying the second version  
6.8 Token streams  
6.9 Program structure  

Chapter 7 Completing a Program  
7.1 Introduction  
7.2 Input and output  
7.3 Error handling  
7.4 Negative numbers  
7.5 Remainder: %  
7.6 Cleaning up the code  
7.7 Recovering from errors  
7.8 Variables  

Chapter 8 Technicalities: Functions, etc.  
8.1 Technicalities  
8.2 Declarations and definitions  
8.3 Header files  
8.4 Scope  
8.5 Function call and return  
8.6 Order of evaluation  
8.7 Namespaces  

Chapter 9 Technicalities: Classes, etc.  
9.1 User-defined types  
9.2 Classes and members  
9.3 Interface and implementation  
9.4 Evolving a class  
9.5 Enumerations  
9.6 Operator overloading
Table of Contents

9.7 Class interfaces
9.8 The Date class

Part II: Input and Output

Chapter 10 Input and Output Streams
  10.1 Input and output
  10.2 The I/O stream model
  10.3 Files
  10.4 Opening a file
  10.5 Reading and writing a file
  10.6 I/O error handling
  10.7 Reading a single value
  10.8 User-defined output operators
  10.9 User-defined input operators
  10.10 A standard input loop
  10.11 Reading a structured file

Chapter 11 Customizing Input and Output
  11.1 Regularity and irregularity
  11.2 Output formatting
  11.3 File opening and positioning
  11.4 String streams
  11.5 Line-oriented input
  11.6 Character classification
  11.7 Using nonstandard separators
  11.8 And there is so much more

Chapter 12 A Display Model
  12.1 Why graphics?
  12.2 A display model
  12.3 A first example
  12.4 Using a GUI library
Table of Contents

12.5 Coordinates
12.6 Shapes
12.7 Using Shape primitives
12.8 Getting this to run

Chapter 13 Graphics Classes
  13.1 Overview of graphics classes
  13.2 Point and Line
  13.3 Lines
  13.4 Color
  13.5 Line_style
  13.6 Open_polyline
  13.7 Closed_polyline
  13.8 Polygon
  13.9 Rectangle
  13.10 Managing unnamed objects
  13.11 Text
  13.12 Circle
  13.13 Ellipse
  13.14 Marked_polyline
  13.15 Marks
  13.16 Mark
  13.17 Images

Chapter 14 Graphics Class Design
  14.1 Design principles
  14.2 Shape
  14.3 Base and derived classes
  14.4 Benefits of object-oriented programming

Chapter 15 Graphing Functions and Data
  15.1 Introduction
Table of Contents

15.2 Graphing simple functions
15.3 Function
15.4 Axis
15.5 Approximation
15.6 Graphing data

Chapter 16 Graphical User Interfaces
16.1 User interface alternatives
16.2 The Next button
16.3 A simple window
16.4 Button and other Widgets
16.5 An example
16.6 Control inversion
16.7 Adding a menu
16.8 Debugging GUI code

Part III: Data and Algorithms
Chapter 17 Vector and Free Store
17.1 Introduction
17.2 vector basics
17.3 Memory, addresses, and pointers
17.4 Free store and pointers
17.5 Destructors
17.6 Access to elements
17.7 Pointers to class objects
17.8 Messing with types: void* and casts
17.9 Pointers and references
17.10 The this pointer

Chapter 18 Vectors and Arrays
18.1 Introduction
18.2 Initialization
Table of Contents

21.7 Copying
21.8 Sorting and searching
21.9 Container algorithms

Part IV: Broadening the View

Chapter 22 Ideals and History
  22.1 History, ideals, and professionalism
  22.2 Programming language history overview

Chapter 23 Text Manipulation
  23.1 Text
  23.2 Strings
  23.3 I/O streams
  23.4 Maps
  23.5 A problem
  23.6 The idea of regular expressions
  23.7 Searching with regular expressions
  23.8 Regular expression syntax
  23.9 Matching with regular expressions
  23.10 References

Chapter 24 Numerics
  24.1 Introduction
  24.2 Size, precision, and overflow
  24.3 Arrays
  24.4 C-style multidimensional arrays
  24.5 The Matrix library
  24.6 An example: solving linear equations
  24.7 Random numbers
  24.8 The standard mathematical functions
  24.9 Complex numbers
  24.10 References
Table of Contents

Chapter 25 Embedded Systems Programming
   25.1 Embedded systems
   25.2 Basic concepts
   25.3 Memory management
   25.4 Addresses, pointers, and arrays
   25.5 Bits, bytes, and words
   25.6 Coding standards

Chapter 26 Testing
   26.1 What we want
   26.2 Proofs
   26.3 Testing
   26.4 Design for testing
   26.5 Debugging
   26.6 Performance
   26.7 References

Chapter 27 The C Programming Language
   27.1 C and C++: siblings
   27.2 Functions
   27.3 Minor language differences
   27.4 Free store
   27.5 C-style strings
   27.6 Input/output: stdio
   27.7 Constants and macros
   27.8 Macros
   27.9 An example: intrusive containers

Part V: Appendices
   Appendix A: Language Summary
      A.1 General
Table of Contents

A.2 Literals
A.3 Identifiers
A.4 Scope, storage class, and lifetime
A.5 Expressions
A.6 Statements
A.7 Declarations
A.8 Built-in types
A.9 Functions
A.10 User-defined types
A.11 Enumerations
A.12 Classes
A.13 Templates
A.14 Exceptions
A.15 Namespaces
A.16 Aliases
A.17 Preprocessor directives

Appendix B: Standard Library Summary
   B.1 Overview
   B.2 Error handling
   B.3 Iterators
   B.4 Containers
   B.5 Algorithms
   B.6 STL utilities
   B.7 I/O streams
   B.8 String manipulation
   B.9 Numerics
   B.10 Time
   B.11 C standard library functions
   B.12 Other libraries
Table of Contents

Appendix C: Getting Started with Visual Studio
   C.1 Getting a program to run
   C.2 Installing Visual Studio
   C.3 Creating and running a program
   C.4 Later

Appendix D: Installing FLTK
   D.1 Introduction
   D.2 Downloading FLTK
   D.3 Installing FLTK
   D.4 Using FLTK in Visual Studio
   D.5 Testing if it all worked

Appendix E: GUI Implementation
   E.1 Callback implementation
   E.2 Widget implementation
   E.3 Window implementation
   E.4 Vector_ref
   E.5 An example: manipulating Widgets

Glossary
   A
   B
   C
   D
   E
   F
   G
   H
   I
Table of Contents

L
M
O
P
R
S
T
U
V
W
Bibliography
Index