

Contents

Preface	v
1 Basic Concepts of C++	1
1.1 Objects, Values, Types, Variables	1
1.1.1 Objects and Addresses	1
1.1.2 Values and Types	2
1.1.3 Referring to Objects and Values: Variables, Constants, Ex- pressions	2
1.2 Defining and Manipulating Objects	3
1.3 References	4
1.4 Functions	6
1.4.1 Defining a Function	6
1.4.2 Calling a Function	7
1.4.3 More on Parameter Passing	10
1.5 Program Structure	12
1.5.1 Translation Units, Definitions and Declarations	12
1.5.2 The Compilation Process	13
1.5.3 Program Organization	15
1.5.4 Linking and Executing a Program	17
1.6 Syntactic and Lexical Considerations	19
1.6.1 Identifiers and Keywords	19
1.6.2 Comments	20
1.7 Scopes and Namespaces	21
1.7.1 Scopes	21
1.7.2 Namespaces	22

2	Built-in Types	25
2.1	Arithmetic Types	25
2.2	Conversions for Arithmetic Types	27
2.3	Arithmetic Type Operations	28
2.4	String Literals	32
3	Functions	37
3.1	Function Declarations	37
3.1.1	Function Types, Signatures	37
3.1.2	Default Arguments	38
3.1.3	Unspecified Number of Arguments	38
3.1.4	Inline Functions	39
3.2	Overloading	41
3.3	Function Definitions	42
3.3.1	Statements	42
3.3.2	Control Flow Statements	43
	The compound statement and the sequence operator	43
	The if statement and the ? operator	43
	The while statements	44
	The for statement	45
	The switch statement	46
	The return, break and continue statements	47
3.3.3	Local and Static Variables	49
	Local object lifetime	49
	Persistent local objects	49
4	User-Defined Types	51
4.1	Abstract Data Types	51
4.2	Classes	53
4.2.1	Class Members, Access Specification	53
4.2.2	Class Objects	54
4.2.3	Data Member Declarations and Object Layout	54
4.2.4	Class Scope	55
4.2.5	Function Member Declaration	55
4.2.6	Overloading Member Functions	56
4.2.7	Initializing a Class Object	56

The default constructor	58
The copy constructor	58
4.2.8 Function Member Definition	59
4.2.9 Inline Function Members	60
4.2.10 Member Functions with Default Parameters	61
4.2.11 User-Defined Conversions	61
4.2.12 Operator Overloading	63
Overloading the assignment operator	65
Overloading the increment and decrement operators	66
Forbidding operators	67
4.2.13 Finalizing an Object	68
4.2.14 Member Objects	69
4.2.15 Friends	71
4.2.16 Nested Classes	72
4.2.17 Static Members	74
4.2.18 Implementation Aspects	74
4.2.19 Example: the Rational Class	76
4.3 Enumerations	80
4.4 Typedef	81
5 Built-in Type Constructors	83
5.1 Constant Objects	83
5.2 Pointers	89
5.2.1 Pointers, Addresses, Dereferencing	89
5.2.2 Handles	90
5.2.3 Member Selection from Pointers	91
5.2.4 Constant Pointers	91
5.2.5 Pointers vs. References	92
5.2.6 The this Pointer	93
5.3 Arrays	95
5.3.1 Initializing an Array	96
5.3.2 Arrays vs. Pointers, Pointer Arithmetic	97
5.3.3 Array Parameters	99
5.3.4 Multidimensional Arrays	101
5.4 Command Line Processing	101

5.5	Pointers to Functions	102
5.6	Pointers to Members	102
5.7	Memory Management	104
5.7.1	Static, Global, Automatic and Member Objects	104
5.7.2	Free Objects	104
5.7.3	Encapsulating Pointers	107
5.8	More Operator Overloading	110
5.8.1	Overloading new , delete	110
5.8.2	Placement New	113
5.8.3	Smart Pointers	114
6	User-Defined Type Constructors	117
6.1	Function Templates	117
6.1.1	Template Argument Deduction	120
6.1.2	Overloading Function Templates	120
6.1.3	Explicit Specializations of Function Templates	121
6.1.4	A Generic Sort Function	122
6.2	Class Templates	123
6.2.1	Friends of Class Templates	126
6.2.2	Nested Class Templates	127
6.2.3	Function Objects	128
6.2.4	Class Template Specialization	131
6.2.5	Reference Counting Pointers	132
6.2.6	Auto Pointers	134
7	Generic Programming Using the STL	137
7.1	Generic Programming	137
7.2	Iterators	140
7.2.1	Types Associated with an Iterator	143
7.2.2	Iterator Traits	146
7.2.3	Dispatching on the Iterator Category	148
7.3	Stream Iterators	149
7.3.1	Input Stream Iterator	149
7.3.2	Output Stream Iterator	151
7.4	STL Containers	152
7.4.1	Pair	153

7.4.2	List	153
7.4.3	Vector	155
7.4.4	Map	157
7.4.5	Set	161
7.4.6	Other Containers	163
7.4.7	Container Adaptors	163
7.5	STL Algorithms	164
7.5.1	Non-mutating Algorithms	164
	Finding elements in a range	165
	Finding subranges	165
	Counting elements in a range	166
	Processing a range	166
	Comparing ranges	167
	Minimum and maximum	168
7.5.2	Basic Mutating Algorithms	169
	Copying ranges	169
	Swapping elements	169
	Transforming a range	169
	Replacing elements in a range	170
	Filling a range	170
	Removing elements	170
	Permuting algorithms	171
	Partitioning ranges	172
	Random shuffling and sampling	172
	Generalized numeric algorithms	173
7.5.3	Sorting and Searching	174
	Sorting ranges	174
	Operations on sorted ranges	175
	Set operations	176
	Heap operations	177
7.6	Iterator Adaptors	179
7.6.1	Insert Iterators	179
7.6.2	Reverse Iterators	181

8.1	Derived Classes	183
8.1.1	Construction and Destruction of Derived Class Objects	185
8.1.2	Inheritance and Scope	185
8.1.3	Inheritance and Conversions	186
8.1.4	Inheritance and Arrays	187
8.2	Virtual Member Functions	187
8.2.1	Implementation of Virtual Member Functions	189
8.2.2	Pure Virtual Function Members and Abstract Classes	191
8.2.3	Virtual Destructors	192
8.3	Derivation and Access Control	193
8.3.1	Protected Members	193
8.3.2	Protected and Private Derivation	194
8.4	Multiple and Virtual Inheritance	195
8.4.1	Multiple Inheritance	195
8.4.2	Implementing Multiple Inheritance	196
8.4.3	Virtual Inheritance	197
8.5	Object-Oriented Programming	199
8.5.1	Class Hierarchies	199
8.5.2	Polymorphism	199
8.6	Run-Time Type Identification	201
9	Exceptions	205
9.1	Throwing and Catching Exceptions	205
9.2	Run-Time Behaviour	208
9.3	Exceptions, Constructors and Destructors	210
9.3.1	Exceptions and Resource Management	210
9.3.2	Constructors Throwing Exceptions	211
9.3.3	Destructors Throwing Exceptions	212
9.4	Exception Specifications	213
9.5	Standard Exceptions	214
10	Iostreams	217
10.1	Requirements	217
10.2	Design	218
10.3	Streambuf	219
10.4	Stream Base Classes	223

10.4.1	Ios_base	223
10.4.2	Basic_ios<CharT,Traits>	225
10.5	Stream Classes	227
10.5.1	Basic_istream<CharT,Traits>	227
10.5.2	Basic_ostream<CharT,Traits>	230
10.5.3	Basic_iostream<CharT,Traits>	231
10.6	Manipulators	231
10.7	File Streams	234
10.7.1	Basic_ifstream<CharT,Traits>	235
10.7.2	Basic_ofstream<CharT,Traits>	236
10.7.3	Basic_fstream<CharT,Traits>	236
10.8	String Streams	237
11	Introduction to Program Design	241
11.1	Motivation: Properties of Good Programs	241
11.2	Abstractions	246
11.3	Criteria For Good Abstractions	252
11.4	The Design Process	259
11.4.1	Finding Abstractions	259
11.4.2	Designing Classes	260
11.4.3	Refactoring	260
11.4.4	Documentation	260
11.4.5	Patterns	261
A	C++ Operators	265
B	The String Class	267
	Bibliography	273
	Index	275