

OSBORNE

# The Complete Reference

Schildt's classic C++  
reference—updated  
and expanded

*Authorized  
Edition For  
Sale in Pakistan  
& Afghanistan*

# C++

## Fourth Edition

Covers the International  
Standard for C++,  
including keywords,  
syntax, and libraries

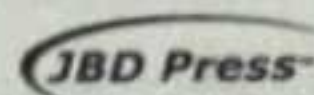
Includes advanced features  
such as overloading, inheritance,  
virtual functions, namespaces,  
templates, the STL, and RTTI

Works with all  
C++ compilers,  
including Visual  
C++

**Herbert Schildt**

FREE  
CODE  
ONLINE

McGraw-Hill/Osborne

 JBD Press

# **C++:** **The Complete Reference,** **Fourth Edition**

**Herbert Schildt**

**Jahangir Book Depot**

**Lahore, Rawalpindi, Multan,**



# Contents at a Glance

---

## Part I The Foundation of C++: The C Subset

■	1	An Overview of C .....	3
■	2	Expressions .....	13
■	3	Statements .....	57
■	4	Arrays and Null-Terminated Strings .....	89
■	5	Pointers .....	113
■	6	Functions .....	137
■	7	Structures, Unions, Enumerations, and User-Defined Types .....	161
■	8	C-Style Console I/O .....	187
■	9	File I/O .....	211
■	10	The Preprocessor and Comments .....	237

## Part II C++

■	11	An Overview of C++ .....	255
■	12	Classes and Objects .....	289

13	Arrays, Pointers, References, and the Dynamic Allocation Operators .....	325
14	Function Overloading, Copy Constructors, and Default Arguments .....	359
15	Operator Overloading .....	369
16	Inheritance .....	417
17	Virtual Functions and Polymorphism .....	443
18	Templates .....	459
19	Exception Handling .....	487
20	The C++ I/O System Basics .....	519
21	C++ File I/O .....	539
22	Run-Time Type ID and the Casting Operators .....	567
23	Namespaces, Conversion Functions, and Other Advanced Topics .....	591
24	Introducing the Standard Template Library .....	629

### Part III The Standard Function Library

25	The C-Based I/O Functions .....	699
26	The String and Character Functions .....	723
27	The Mathematical Functions .....	737
28	Time, Date, and Localization Functions .....	747
29	The Dynamic Allocation Functions .....	757
30	Utility Functions .....	761
31	The Wide-Character Functions .....	775

### Part IV The Standard C++ Class Library

32	The Standard C++ I/O Classes .....	787
33	The STL Container Classes .....	811
34	The STL Algorithms .....	839
35	STL Iterators, Allocators, and Function Objects .....	861
36	The String Class .....	881
37	The Numeric Classes .....	897
38	Exception Handling and Miscellaneous Classes .....	925



**Part V Applying C++**

<b>39</b>	Integrating New Classes: A Custom String Class .....	935
<b>40</b>	Parsing Expressions .....	963
<b>A</b>	The .NET Managed Extensions to C++ .....	999
<b>B</b>	C++ and the Robotics Age .....	1005
	Index. ....	1009

# Contents

---

Introduction .....	xxix
--------------------	------

## Part I

### The Foundation of C++: The C Subset

<b>1</b>	An Overview of C .....	3
	The Origins and History of C .....	4
	C Is a Middle-Level Language .....	5
	C Is a Structured Language .....	6
	C Is a Programmer's Language .....	8
	The Form of a C Program .....	9
	The Library and Linking .....	10
	Separate Compilation .....	12
	Understanding the .C and .CPP File Extensions .....	12

<b>2</b>	Expressions .....	13
	The Five Basic Data Types .....	14
	Modifying the Basic Types .....	15
	Identifier Names .....	16
	Variables .....	17
	Where Variables Are Declared .....	18
	Local Variables .....	18



Formal Parameters .....	
Global Variables .....	
The const and volatile Qualifiers .....	
const .....	
volatile .....	
Storage Class Specifiers .....	
extern .....	
static Variables .....	
register Variables .....	
Variable Initializations .....	
Constants .....	
Hexadecimal and Octal Constants .....	
String Constants .....	
Backslash Character Constants .....	
Operators .....	
The Assignment Operator .....	
Type Conversion in Assignments .....	
Multiple Assignments .....	
Arithmetic Operators .....	
Increment and Decrement .....	
Relational and Logical Operators .....	
Bitwise Operators .....	
The ? Operator .....	
The & and * Pointer Operators .....	
The Compile-Time Operator sizeof .....	
The Comma Operator .....	
The Dot (.) and Arrow (->) Operators .....	
The [ ] and ( ) Operators .....	
Precedence Summary .....	
Expressions .....	
Order of Evaluation .....	
Type Conversion in Expressions .....	
Casts .....	
Spacing and Parentheses .....	
Compound Assignments .....	

<b>3</b> Statements .....	
True and False in C and C++ .....	
Selection Statements .....	
if .....	
Nested ifs .....	
The if-else-if Ladder .....	
The ? Alternative .....	
The Conditional Expression .....	



switch .....	67
Nested switch Statements .....	70
Iteration Statements .....	70
The for Loop .....	70
for Loop Variations .....	72
The Infinite Loop .....	76
for Loops with No Bodies .....	77
The while Loop .....	77
The do-while Loop .....	79
Declaring Variables Within Selection and Iteration Statements ..	81
Jump Statements .....	82
The return Statement .....	82
The goto Statement .....	83
The break Statement .....	83
The exit( ) Function .....	85
The continue Statement .....	86
Expression Statements .....	88
Block Statements .....	88
<b>4 Arrays and Null-Terminated Strings .....</b>	<b>89</b>
Single-Dimension Arrays .....	90
Generating a Pointer to an Array .....	92
Passing Single-Dimension Arrays to Functions .....	92
Null-Terminated Strings .....	94
Two-Dimensional Arrays .....	96
Arrays of Strings .....	100
Multidimensional Arrays .....	101
Indexing Pointers .....	102
Array Initialization .....	105
Unsize Array Initializations .....	106
A Tic-Tac-Toe Example .....	108
<b>5 Pointers .....</b>	<b>113</b>
What Are Pointers? .....	114
Pointer Variables .....	115
The Pointer Operators .....	115
Pointer Expressions .....	116
Pointer Assignments .....	117
Pointer Arithmetic .....	117
Pointer Comparisons .....	119
Pointers and Arrays .....	121
Arrays of Pointers .....	122
Multiple Indirection .....	123
Initializing Pointers .....	125
Pointers to Functions .....	126



C's Dynamic Allocation Functions .....	
Problems with Pointers .....	

<b>6</b>	<b>Functions .....</b>	
	The General Form of a Function .....	
	Scope Rules of Functions .....	
	Function Arguments .....	
	Call by Value, Call by Reference .....	
	Creating a Call by Reference .....	
	Calling Functions with Arrays .....	
	argc and argv—Arguments to main( ) .....	
	The return Statement .....	
	Returning from a Function .....	
	Returning Values .....	
	Returning Pointers .....	
	Functions of Type void .....	
	What Does main( ) Return? .....	
	Recursion .....	
	Function Prototypes .....	
	Standard Library Function Prototypes .....	
	Declaring Variable-Length Parameter Lists .....	
	Old-Style Versus Modern FunctionParameter Declarations .....	

<b>7</b>	<b>Structures, Unions, Enumerations, and User-Defined Types .....</b>	
	Structures .....	
	Accessing Structure Members .....	
	Structure Assignments .....	
	Arrays of Structures .....	
	Passing Structures to Functions .....	
	Passing Structure Members to Functions .....	
	Passing Entire Structures to Functions .....	
	Structure Pointers .....	
	Declaring a Structure Pointer .....	
	Using Structure Pointers .....	
	Arrays and Structures Within Structures .....	
	Bit-Fields .....	
	Unions .....	
	Enumerations .....	
	Using sizeof to Ensure Portability .....	
	typedef .....	

<b>8</b>	<b>C-Style Console I/O .....</b>	
	An Important Application: M.....	



Reading and Writing Characters .....	189
A Problem with <code>getchar()</code> .....	190
Alternatives to <code>getchar()</code> .....	190
Reading and Writing Strings .....	192
Formatted Console I/O .....	195
<code>printf()</code> .....	195
Printing Characters .....	196
Printing Numbers .....	196
Displaying an Address .....	198
The <code>%n</code> Specifier .....	198
Format Modifiers .....	199
The Minimum Field Width Specifier .....	199
The Precision Specifier .....	200
Justifying Output .....	201
Handling Other Data Types .....	202
The <code>*</code> and <code>#</code> Modifiers .....	202
<code>scanf()</code> .....	203
Format Specifiers .....	203
Inputting Numbers .....	203
Inputting Unsigned Integers .....	205
Reading Individual Characters Using <code>scanf()</code> .....	205
Reading Strings .....	205
Inputting an Address .....	206
The <code>%n</code> Specifier .....	206
Using a Scanset .....	206
Discarding Unwanted White Space .....	207
Non-White-Space Characters in the Control String .....	208
You Must Pass <code>scanf()</code> Addresses .....	208
Format Modifiers .....	208
Suppressing Input .....	209
<b>9 File I/O .....</b>	<b>211</b>
C Versus C++ File I/O .....	212
Streams and Files .....	212
Streams .....	212
Text Streams .....	213
Binary Streams .....	213
Files .....	213
File System Basics .....	214
The File Pointer .....	215
Opening a File .....	215
Closing a File .....	217
Writing a Character .....	217
Reading a Character .....	218



Using fopen( ), getc( ), putc( ), and fclose( )	237
Using feof( )	238
Working with Strings: fputs( ) and fgets( )	238
rewind( )	240
ferror( )	241
Erasing Files	242
Flushing a Stream	242
fread( ) and fwrite( )	243
Using fread( ) and fwrite( )	245
fseek( ) and Random-Access I/O	246
fprintf( ) and fscanf( )	247
The Standard Streams	248
The Console I/O Connection	250
Using freopen( ) to Redirect the Standard Streams	252
<b>10 The Preprocessor and Comments</b>	<b>253</b>
The Preprocessor	253
#define	253
Defining Function-like Macros	254
#error	255
#include	255
Conditional Compilation Directives	256
#if, #else, #elif, and #endif	257
#ifdef and #ifndef	258
#undef	259
Using defined	259
#line	260
#pragma	260
The # and ## Preprocessor Operators	261
Predefined Macro Names	261
Comments	262
Single-Line Comments	262

## Part II

### C++

<b>11 An Overview of C++</b>	<b>255</b>
The Origins of C++	256
What Is Object-Oriented Programming?	257
Encapsulation	258
Polymorphism	259
Inheritance	259
Some C++ Fundamentals	260
A Sample C++ Program	260



A Closer Look at the I/O Operators .....	263
Declaring Local Variables .....	264
No Default to int .....	265
The bool Data Type .....	266
Old-Style vs. Modern C++ .....	267
The New C++ Headers .....	268
Namespaces .....	269
Working with an Old Compiler .....	270
Introducing C++ Classes .....	270
Function Overloading .....	275
Operator Overloading .....	278
Inheritance .....	278
Constructors and Destructors .....	283
The C++ Keywords .....	287
The General Form of a C++ Program .....	288

<b>12</b> Classes and Objects .....	289
Classes .....	290
Structures and Classes Are Related .....	293
Unions and Classes Are Related .....	295
Anonymous Unions .....	296
Friend Functions .....	297
Friend Classes .....	302
Inline Functions .....	303
Defining Inline Functions Within a Class .....	306
Parameterized Constructors .....	307
Constructors with One Parameter: A Special Case .....	309
Static Class Members .....	310
Static Data Members .....	310
Static Member Functions .....	315
When Constructors and Destructors Are Executed .....	317
The Scope Resolution Operator .....	319
Nested Classes .....	319
Local Classes .....	320
Passing Objects to Functions .....	320
Returning Objects .....	323
Object Assignment .....	324

<b>13</b> Arrays, Pointers, References, and the Dynamic Allocation Operators .....	325
Arrays of Objects .....	326
Creating Initialized vs. Uninitialized Arrays .....	328
Pointers to Objects .....	329
Type Checking C++ Pointers .....	332



The this Pointer .....	
Pointers to Derived Types .....	
Pointers to Class Members .....	
References .....	
Reference Parameters .....	
Passing References to Objects .....	
Returning References .....	
Independent References .....	
References to Derived Types .....	
Restrictions to References .....	
A Matter of Style .....	
C++'s Dynamic Allocation Operators .....	
Initializing Allocated Memory .....	
Allocating Arrays .....	
Allocating Objects .....	
The nothrow Alternative .....	
The Placement Form of new .....	

## 14 Function Overloading, Copy Constructors, and Default Arguments .....

Function Overloading .....	
Overloading Constructors .....	
Overloading a Constructor to Gain Flexibility .....	
Allowing Both Initialized and Uninitialized Objects .....	
Copy Constructors .....	
Finding the Address of an Overloaded Function .....	
The overload Anachronism .....	
Default Function Arguments .....	
Default Arguments vs. Overloading .....	
Using Default Arguments Correctly .....	
Function Overloading and Ambiguity .....	

## 15 Operator Overloading .....

Creating a Member Operator Function .....	
Creating Prefix and Postfix Forms .....	
of the Increment and Decrement Operators .....	
Overloading the Shorthand Operators .....	
Operator Overloading Restrictions .....	
Operator Overloading Using a Friend Function .....	
Using a Friend to Overload ++ or -- .....	
Friend Operator Functions Add Flexibility .....	
Overloading new and delete .....	
Overloading new and delete for Arrays .....	
Overloading the nothrow Version of new and delete .....	



Overloading Some Special Operators .....	407
Overloading [ ] .....	407
Overloading ( ) .....	411
Overloading -> .....	413
Overloading the Comma Operator .....	414
<b>16 Inheritance</b> .....	<b>417</b>
Base-Class Access Control .....	418
Inheritance and protected Members .....	420
Protected Base-Class Inheritance .....	424
Inheriting Multiple Base Classes .....	425
Constructors, Destructors, and Inheritance .....	426
When Constructors and Destructors Are Executed .....	426
Passing Parameters to Base-Class Constructors .....	430
Granting Access .....	434
Virtual Base Classes .....	437
<b>17 Virtual Functions and Polymorphism</b> .....	<b>443</b>
Virtual Functions .....	444
Calling a Virtual Function Through a Base Class Reference .....	447
The Virtual Attribute Is Inherited .....	448
Virtual Functions Are Hierarchical .....	450
Pure Virtual Functions .....	453
Abstract Classes .....	455
Using Virtual Functions .....	455
Early vs. Late Binding .....	458
<b>18 Templates</b> .....	<b>459</b>
Generic Functions .....	460
A Function with Two Generic Types .....	463
Explicitly Overloading a Generic Function .....	463
Overloading a Function Template .....	466
Using Standard Parameters with Template Functions .....	466
Generic Function Restrictions .....	467
Applying Generic Functions .....	468
A Generic Sort .....	469
Compacting an Array .....	470
Generic Classes .....	472
An Example with Two Generic Data Types .....	476
Applying Template Classes: A Generic Array Class .....	477
Using Non-Type Arguments with Generic Classes .....	479
Using Default Arguments with Template Classes .....	481
Explicit Class Specializations .....	483



	The typename and export Keywords .....	60
	The Power of Templates .....	61
<b>19</b>	<b>Exception Handling .....</b>	<b>62</b>
	Exception Handling Fundamentals .....	62
	Catching Class Types .....	62
	Using Multiple catch Statements .....	62
	Handling Derived-Class Exceptions .....	62
	Exception Handling Options .....	62
	Catching All Exceptions .....	62
	Restricting Exceptions .....	62
	Rethrowing an Exception .....	62
	Understanding terminate( ) and unexpected( ) .....	62
	Setting the Terminate and Unexpected Handlers .....	62
	The uncaught_exception( ) Function .....	62
	The exception and bad_exception Classes .....	62
	Applying Exception Handling .....	62
<b>20</b>	<b>The C++ I/O System Basics .....</b>	<b>62</b>
	Old vs. Modern C++ I/O .....	62
	C++ Streams .....	62
	The C++ Stream Classes .....	62
	C++'s Predefined Streams .....	62
	Formatted I/O .....	62
	Formatting Using the ios Members .....	62
	Setting the Format Flags .....	62
	Clearing Format Flags .....	62
	An Overloaded Form of setf( ) .....	62
	Examining the Formatting Flags .....	62
	Setting All Flags .....	62
	Using width( ), precision( ), and fill( ) .....	62
	Using Manipulators to Format I/O .....	62
	Overloading << and >> .....	62
	Creating Your Own Inserters .....	62
	Creating Your Own Extractors .....	62
	Creating Your Own Manipulator Functions .....	62
<b>21</b>	<b>C++ File I/O .....</b>	<b>53</b>
	<fstream> and the File Classes .....	54
	Opening and Closing a File .....	54
	Reading and Writing Text Files .....	54
	Unformatted and Binary I/O .....	54
	Characters vs. Bytes .....	54
	put( ) and get( ) .....	54
	read( ) and write( ) .....	54



More get( ) Functions .....	551
getline( ) .....	551
Detecting EOF .....	553
The ignore( ) Function .....	555
peek( ) and putback( ) .....	556
flush( ) .....	556
Random Access .....	557
Obtaining the Current File Position .....	561
I/O Status .....	561
Customized I/O and Files .....	563
<b>22 Run-Time Type ID and the Casting Operators .....</b>	<b>567</b>
Run-Time Type Identification (RTTI) .....	568
The Casting Operators .....	578
dynamic_cast .....	578
<b>23 Namespaces, Conversion Functions,     and Other Advanced Topics .....</b>	<b>591</b>
Namespaces .....	592
Namespace Fundamentals .....	592
using .....	596
Unnamed Namespaces .....	598
Some Namespace Options .....	599
The std Namespace .....	601
Creating Conversion Functions .....	603
const Member Functions and mutable .....	607
Volatile Member Functions .....	609
Explicit Constructors .....	610
The Member Initialization Syntax .....	611
Using the asm Keyword .....	616
Linkage Specification .....	617
Array-Based I/O .....	618
The Array-Based Classes .....	619
Creating an Array-Based Output Stream .....	619
Using an Array as Input .....	621
Input/Output Array-Based Streams .....	623
Using Dynamic Arrays .....	624
Using Binary I/O with Array-Based Streams .....	625
Using Binary I/O with Array-Based Streams .....	626
Summarizing the Differences Between C and C++ .....	626
<b>24 Introducing the Standard Template Library .....</b>	<b>629</b>
An Overview of the STL .....	630
Containers .....	630



Algorithms .....	
Iterators .....	
Other STL Elements .....	
The Container Classes .....	
General Theory of Operation .....	
Vectors .....	
Accessing a Vector Through an Iterator .....	
Inserting and Deleting Elements in a Vector .....	
Storing Class Objects in a Vector .....	
Lists .....	
Understanding end( ) .....	
push_front( ) vs. push_back( ) .....	
Sort a List .....	
Merging One List with Another .....	
Storing Class Objects in a List .....	
Maps .....	
Storing Class Objects in a Map .....	
Algorithms .....	
Counting .....	
Removing and Replacing Elements .....	
Reversing a Sequence .....	
Transforming a Sequence .....	
Using Function Objects .....	
Unary and Binary Function Objects .....	
Using the Built-in Function Objects .....	
Creating a Function Object .....	
Using Binders .....	
The string Class .....	
Some string Member Functions .....	
Strings Are Containers .....	
Putting Strings into Other Containers .....	
Final Thoughts on the STL .....	

### Part III

## The Standard Function Library

### 25 The C-Based I/O Functions

clearerr .....	699
fclose .....	700
feof .....	701
ferror .....	701
fflush .....	701
fgetc .....	702
fgetpos .....	702

fgets	703
fopen	703
fprintf	705
fputc	705
fputs	706
fread	706
freopen	706
fscanf	707
fseek	707
fsetpos	708
ftell	708
fwrite	709
getc	709
getchar	710
gets	710
perror	710
printf	711
putc	714
putchar	714
puts	714
remove	715
rename	715
rewind	715
scanf	715
setbuf	719
setvbuf	719
sprintf	720
sscanf	720
tmpfile	720
tmpnam	721
ungetc	721
vprintf, vfprintf, and vsprintf	722

<b>26</b>	<b>The String and Character Functions</b>	<b>723</b>
	isalnum	724
	isalpha	724
	iscntrl	725
	isdigit	725
	isgraph	725
	islower	725
	isprint	726
	ispunct	726
	isspace	726
	isupper	