

PROGRAMMING IN C++

PAUL M. CHIRLIAN

COMPUTER SCIENCE COMPUTER SCIENCE COMPUTER SCIENCE COMPUTER



MAXWELL
MACMILLAN
INTERNATIONAL
EDITIONS

COMPUTER SCIENCE COMPUTER SCIENCE COMPUTER SCIENCE COMPUTER



PROGRAMMING IN C++

Paul M. Chirlian

Stevens Institute of Technology

452
SEMINAR LIBRARY
Department of Computer Science
UNIVERSITY OF KARACHI

5-4-95



Contents

CHAPTER 1 **An Introduction to C++**

1-1	Some Fundamentals of C++ Programming	1
1-2	Some Simple C++ Programs	4
1-3	Running C++ Programs	11
1-4	A First Discussion of Programming Errors	12

CHAPTER 2 **Arithmetic Operations**

2-1	Floating-Point Operations	15
2-2	Integer Operations	18
2-3	Assignment	21
2-4	Hierarchy—Use of Parentheses	25
2-5	Mixed-Mode Operations	27
2-6	Initial Values—Constants	31
2-7	Library Functions	35
2-8	Complex Numbers	38

CHAPTER 3 **Some Fundamentals of C++ Programming**

3-1 Characters and Strings	43
3-2 Streams—Basic Input and Output	48
3-3 A First Discussion of Scope and Lifetime—Blocks	53
3-4 Pointers	56
3-5 Enumerated Constants—Elementary User-Defined Types	64

CHAPTER 4 **Program Control**

4-1 Logical Operators	67
4-2 The if Construction	73
4-3 The if-else Construction	77
4-4 Looping	80
4-5 Nested Loops	87
4-6 The switch-case Construction	90
4-7 Bitwise Operations	93
4-8 Hierarchy—Some Additional Topics	96
4-9 The goto Construction	99
4-10 A Basic Discussion of Structured Programming	102

CHAPTER 5 **Functions**

5-1 Functions	107
5-2 Argument Passing: Pass-by-Value—Pass-by-Reference	112
5-3 Variables	116

5-4	Scope	122
5-5	Libraries	125
5-6	Recursion	127
5-7	Default Arguments—Variable Number of Arguments	132
5-8	Overloaded Functions	137
5-9	Pointers to Functions	139
5-10	Macros and Inline Functions	143
5-11	Structured Programming—Modularization	147

CHAPTER 6 Arrays

6-1	Vectors—One Dimensional Arrays	151
6-2	The Relationship between Vectors and Pointers	155
6-3	Using Vectors with Functions	158
6-4	Multidimensional Arrays	161
6-5	The Relationship between Multidimensional Arrays and Pointers	166
6-6	Evaluation Order	171
6-7	Strings	172
6-8	Command-Line Arguments	184

CHAPTER 7 Elementary Structures and Other User-Defined Data Types

7-1	Structures	189
7-2	Structures, Pointers, and Functions	192

CHAPTER 8 Classes

8-1	Classes	203
8-2	Arrays of Classes—Constructors and Destructors— Assignment	204
8-3	this: A Pointer	214
8-4	Friends	222
8-5	Static Field Variables	226
8-6	Classes and Unions	230
8-7	operator Overloading	233
8-8	The operator[] Function—Subscripting	236
8-9	The operator() Function	242
8-10	User-Defined Type Conversion	246

CHAPTER 9 Derived Classes

9-1	Derived Classes	248
9-2	Class Trees—Class Hierarchies	253
9-3	Pointers and Derived Classes	253
9-4	virtual Functions	262
9-5	Generic Classes	264
9-6	Linked Lists: Some General Comments	267

CHAPTER 10 Streams

10-1	Output Streams—User-Defined Types— Output Redirection	270
------	--	-----

10-2	Input Streams—User-Defined Types— Input Redirection	285
10-3	Stream States	289
10-4	Files	291
10-5	Compiler Directives	297

Index	301
-------	-----



CHAPTER 1

An Introduction to C++

This textbook will show you how to write programs in the C++ programming language. C++ is an outgrowth of the C programming language. Indeed, C programs will run under C++. However, in this book we assume that the reader does not have any knowledge of C, or of any other programming language. We shall start by discussing some simple ideas of programming and then build up to more complex ones. Although the book is written without assuming that the reader knows C, experienced C programmers can use it as well to learn C++. Because C++ is thoroughly discussed, the book also serves as a reference.

C++ is a very versatile language. It incorporates features of object-oriented languages as well as the most modern features of modern programming languages. All aspects of C++ are discussed in this book.

1-1 ■ SOME FUNDAMENTALS OF C++ PROGRAMMING

The operations of a computer are directed by a set of instructions called program. The program that actually directs a computer is written in *machine language*. That is, it consists of a sequence of the actual instructions to the computer and the data on which the program is to operate. These are all in binary form, i.e., a sequence of zeros and ones. Machine language programs are very tedious to work with, and because of this, most programs are written in higher-level programming languages. C++ is such a language. For instance, a machine language program that simply multiplies two numbers would consist of a long sequence of instructions having no apparent relationship to multiplication. The same instruction written in C++ would be understandable. For instance, a C++ expression that corresponds to the arithmetic expression