

C++ for Java Programmers

Timothy A. Budd
Oregon State University
Corvallis, Oregon
USA All rights reserved.

No part of this publication may be reproduced,
stored in a retrieval system, or transmitted,
in any form or by any means, electronic, mechanical,
photocopying, recording, or otherwise,
without the prior written permission of the Author.
Copyright 1998 by Timothy A. Budd

December 18, 1998

0

This is not a blank page.

Contents

1	Basic Philosophical Differences	11
1.1	The Language C	11
1.2	The Development of C++	12
1.3	The Legacy Problem	15
1.4	The Language Java	15
1.5	The Better Language	17
1.6	Further Reading	17
2	Fundamental Data Types	21
2.1	Integers	21
2.1.1	Characters	23
2.1.2	Booleans	23
2.1.3	Bit Fields	25
2.2	Floating Point Values	25
2.3	Enumerated Values	26
2.4	The void type	27
2.5	Arrays	27
2.6	Structures and Unions	28
2.7	Object Values	29
2.8	Functions	30
2.8.1	Order of Argument Evaluation	31
2.8.2	The Function Main	32
2.8.3	Alternative Main Entry Points	32
3	Pointers and References	35
3.1	Java Really Does Have Pointers	35
3.2	Pointers on Pointers	36
3.3	The address-of operator	38
3.4	Pointers to Simple Values	38
3.4.1	Pointers to Pointer	40

3.4.2	Pointers and const	40
3.4.3	void * pointers	41
3.4.4	Pointers to Functions	41
3.5	Pointers to Structures	42
3.6	Pointers to Arrays	43
3.7	References	44
3.7.1	Pass by Reference Parameters	46
3.7.2	References as Results	47
4	Memory Management	51
4.1	The Memory Model	52
4.2	Stack Resident Memory Values	52
4.2.1	Lifetime errors	53
4.2.2	Size errors – the Slicing Problem	54
4.3	Heap Resident Memory Values	58
4.3.1	Encapsulating Memory Management	61
4.3.2	Reference Counts	66
5	The Class Definition	71
5.1	Obvious Similarities	71
5.2	Separation of Class and Implementation	72
5.2.1	Interface and Implementation Files	73
5.2.2	The inline directive	74
5.2.3	Prototypes	74
5.2.4	External Declarations	75
5.3	Forward References	76
5.4	Constructors and Initialization	77
5.4.1	Default and Copy constructors	78
5.4.2	Initializers	79
5.4.3	Order of Initialization	81
5.4.4	Combining Constructors	83
5.5	The Orthodox Canonical Class Form	85
5.6	Visibility modifiers	86
5.7	Inner classes versus Nested Classes	87
5.8	Static Initialization	90
5.9	Final Classes	93
6	Polymorphism	97
6.1	Virtual and Non-virtual Overriding	100
6.1.1	Impact of Virtual on Size	101
6.1.2	Obtaining Type Information from a dynamic value	102
6.2	Abstract Classes	102

6.3	Downcasting (Reverse Polymorphism)	103
6.3.1	Simulating The Dynamic Cast	105
6.4	Name Resolution	107
6.5	A Forest, not a Tree	109
6.6	Virtual Destructors	109
6.7	Private Inheritance	110
6.8	Inheritance and Arrays	112
6.9	Overloading	113
7	Operator Overloading	115
7.1	Overloaded Functions or Methods	116
7.2	The Simple Binary Operators	118
7.3	The Comparison Operators	119
7.4	The Increment and Decrement Operators	119
7.5	The Shift Operators	121
7.6	The Assignment Operator	122
7.7	The Compound Assignment Operators	125
7.8	The Subscript Operator	125
7.9	The Parenthesis Operator	126
7.10	The Address Of Operator	128
7.11	The Logical Connectives	129
7.12	The Comma Operator	129
7.13	The Arrow Operator	131
7.14	Pointers to Members and Pointers to Member Functions	132
7.15	Conversion Operators	132
7.16	Memory Management Operators	133
7.17	Disallowing Operations	133
7.18	Implicit Functions and Invocations	133
7.18.1	Implicitly Created Operations	134
7.18.2	Implicit Function Invocations	136
8	Characters and Strings	141
8.1	Characters and Literals Strings	141
8.1.1	Character Literals and the <code>cctype</code> library	141
8.1.2	String literals	142
8.1.3	The <code>cstring</code> library	143
8.2	Constant and Mutable values	144
8.3	The <code>string</code> data type	145
8.4	Example Program – Split a Line into Words	147

9	Templates and Containers	151
9.1	Template Classes	151
9.1.1	Template Methods	154
9.2	Template Functions	154
9.3	The Standard Template Library	156
9.3.1	Containers	156
9.3.2	Iterators	160
9.3.3	Generic Algorithms	164
9.3.4	Function Objects	164
10	Input Output	169
10.1	The <code>stdio</code> library	169
10.1.1	Formatted Output	171
10.2	The Stream I/O facility	173
10.2.1	Stream Input	177
10.2.2	String Streams	177
10.2.3	File Streams	178
10.3	An Example Program	179
11	Exception Handling	183
11.1	Flags and Return Codes	183
11.2	The Assertion Library	185
11.3	The <code>setjmp</code> and <code>longjmp</code> facility	186
11.4	Signals	188
11.5	Exception Handling	189
11.5.1	Exceptions can be Any Type	189
11.5.2	Rethrowing Exceptions	190
11.5.3	No Finally Clause	191
11.5.4	Use References for Exceptions	192
11.5.5	Exception Classes must be Clonable	192
11.5.6	Exceptions Need Not be Documented	193
11.5.7	Standard Exceptions	195
12	Features found only in C++	197
12.1	Global Variables	197
12.2	The Preprocessor	199
12.3	Typedef	201
12.4	The <code>const</code> keyword	202
12.5	Default Arguments	203
12.6	Friends	204
12.7	Name Spaces	206
12.8	Multiple Inheritance	207

12.9 Goto Statements	210
13 Features found only in Java	213
13.1 Wrapper Classes	213
13.2 Interfaces	213
13.3 Inline Classes	214
13.4 Threads	215
13.5 Reflection	215
14 Case Study – Fractions	217
14.1 Classes	218
14.1.1 Interface and Implementation	220
14.2 Constructors	220
14.3 Behavior	222
14.3.1 Member Functions	222
14.3.2 Operators	222
14.3.3 Increment and Decrement	225
14.3.4 Functions	226
14.3.5 Member Function Operators	227
14.3.6 Conversion Operations	229
14.4 Input and Output Streams	230
14.4.1 Stream Input	231
15 Case study – Containers	235
15.1 Graph Shortest Path Problem	235
15.1.1 Shortest Path Algorithm	238
15.1.2 Developing the Data Structures	238
15.2 A Concordance	242
16 Case study – A Card Game	245
16.1 The Class Card	245
16.2 Data and View Classes	247
16.3 The Game	248
16.4 Card Piles–Inheritance in Action	250
16.4.1 The Default Card Pile	253
16.4.2 The Suit Piles	254
16.4.3 The Deck Pile	255
16.4.4 The Discard Pile	257
16.4.5 The Tableau Piles	259
16.5 Playing the Polymorphic Game	261
16.6 The Graphical User Interface	263

17 Case Study: Combining Separate Classes	269
A Include Files	275
A.1 Input and Output	275
A.2 Diagnostics	276
A.3 Strings	276
A.4 STL Containers	276
A.5 STL Utility Routines	277
A.6 Numerical Support	277
A.7 Run-time Support	278
A.8 Miscellaneous	278
Index	283