

---

# Contents

---

<b>Preface</b>	<b>xxx</b>
<b>1 Introduction to Computers, the Internet and the World Wide Web</b>	<b>1</b>
1.1 Introduction	2
1.2 What Is a Computer?	4
1.3 Computer Organization	4
1.4 Evolution of Operating Systems	5
1.5 Personal Computing, Distributed Computing and Client/Server Computing	6
1.6 Machine Languages, Assembly Languages and High-level Languages	7
1.7 The History of C	8
1.8 The C Standard Library	9
1.9 The Key Software Trend: Object Technology	10
1.10 C++ and <i>C++ How to Program</i>	11
1.11 Java and <i>Java How to Program</i>	12
1.12 Other High-level Languages	13
1.13 Structured Programming	13
1.14 The Basics of a typical C Program Development Environment	14
1.15 Hardware Trends	16
1.16 History of the Internet	17
1.17 History of the World Wide Web	18
1.18 General Notes About C and this Book	19
<b>2 Introduction to C Programming</b>	<b>26</b>
2.1 Introduction	27
2.2 A Simple C Program: Printing a Line of Text	27
2.3 Another Simple C Program: Adding Two Integers	31
2.4 Memory Concepts	35

©1994-2000 by Deitel & Associates, Inc. and Prentice Hall. All rights reserved.

2.5	Arithmetic in C	37
2.6	Decision Making: Equality and Relational Operators	40
<b>3</b>	<b>Structured Program Development in C</b>	<b>56</b>
3.1	Introduction	57
3.2	Algorithms	57
3.3	Pseudocode	58
3.4	Control Structures	58
3.5	The <b>if</b> Selection Structure	60
3.6	The <b>if/else</b> Selection Structure	62
3.7	The <b>while</b> Repetition Structure	66
3.8	Formulating Algorithms: Case Study 1 (Counter-Controlled Repetition)	67
3.9	Formulating Algorithms with Top-down, Stepwise Refinement: Case Study 2 (Sentinel-Controlled Repetition)	69
3.10	Formulating Algorithms with Top-down, Stepwise Refinement: Case Study 3 (Nested Control Structures)	75
3.11	Assignment Operators	77
3.12	Increment and Decrement Operators	80
<b>4</b>	<b>C Program Control</b>	<b>100</b>
4.1	Introduction	101
4.2	The Essentials of Repetition	101
4.3	Counter-Controlled Repetition	102
4.4	The <b>for</b> Repetition Structure	104
4.5	The <b>for</b> Structure: Notes and Observations	106
4.6	Examples Using the <b>for</b> Structure	107
4.7	The <b>switch</b> Multiple-Selection Structure	111
4.8	The <b>do/while</b> Repetition Structure	116
4.9	The <b>break</b> and <b>continue</b> Statements	118
4.10	Logical Operators	120
4.11	Confusing Equality (==) and Assignment (=) Operators	122
4.12	Structured Programming Summary	124
<b>5</b>	<b>C Functions</b>	<b>142</b>
5.1	Introduction	143
5.2	Program Modules in C	143
5.3	Math Library Functions	145
5.4	Functions	146
5.5	Function Definitions	147
5.6	Function Prototypes	151
5.7	Header Files	153
5.8	Calling Functions: Call by Value and Call by Reference	154
5.9	Random Number Generation	155
5.10	Example: A Game of Chance	159
5.11	Storage Classes	162
5.12	Scope Rules	164
5.13	Recursion	167

5.14	Example Using Recursion: The Fibonacci Series	170
5.15	Recursion vs. Iteration	174
<b>6</b>	<b>C Arrays</b>	<b>196</b>
6.1	Introduction	197
6.2	Arrays	197
6.3	Declaring Arrays	199
6.4	Examples Using Arrays	199
6.5	Passing Arrays to Functions	211
6.6	Sorting Arrays	216
6.7	Case Study: Computing Mean, Median and Mode Using Arrays	218
6.8	Searching Arrays	221
6.9	Multiple-Subscripted Arrays	226
<b>7</b>	<b>C Pointers</b>	<b>249</b>
7.1	Introduction	250
7.2	Pointer Variable Declarations and Initialization	250
7.3	Pointer Operators	251
7.4	Calling Functions by Reference	254
7.5	Using the <b>const</b> Qualifier with Pointers	258
7.6	Bubble Sort Using Call by Reference	264
7.7	Pointer Expressions and Pointer Arithmetic	269
7.8	The Relationship between Pointers and Arrays	272
7.9	Arrays of Pointers	276
7.10	Case Study: A Card Shuffling and Dealing Simulation	277
7.11	Pointers to Functions	281
<b>8</b>	<b>C Characters and Strings</b>	<b>306</b>
8.1	Introduction	307
8.2	Fundamentals of Strings and Characters	307
8.3	Character Handling Library	309
8.4	String Conversion Functions	314
8.5	Standard Input/Output Library Functions	319
8.6	String Manipulation Functions of the String Handling Library	323
8.7	Comparison Functions of the String Handling Library	325
8.8	Search Functions of the String Handling Library	328
8.9	Memory Functions of the String Handling Library	333
8.10	Other Functions of the String Handling Library	337
<b>9</b>	<b>C Formatted Input/Output</b>	<b>352</b>
9.1	Introduction	353
9.2	Streams	353
9.3	Formatting Output with <b>printf</b>	354
9.4	Printing Integers	354
9.5	Printing Floating-Point Numbers	356
9.6	Printing Strings and Characters	357
9.7	Other Conversion Specifiers	358

9.8	Printing with Field Widths and Precisions	360
9.9	Using Flags in the <code>printf</code> Format-Control String	362
9.10	Printing Literals and Escape Sequences	365
9.11	Formatting Input with <code>scanf</code>	366
<b>10</b>	<b>C Structures, Unions, Bit Manipulations and Enumerations</b>	<b>380</b>
10.1	Introduction	381
10.2	Structure Definitions	381
10.3	Initializing Structures	384
10.4	Accessing Members of Structures	384
10.5	Using Structures with Functions	386
10.6	<code>typedef</code>	386
10.7	Example: High-Performance Card Shuffling and Dealing Simulation	387
10.8	Unions	389
10.9	Bitwise Operators	392
10.10	Bit Fields	400
10.11	Enumeration Constants	403
<b>11</b>	<b>C File Processing</b>	<b>415</b>
11.1	Introduction	416
11.2	The Data Hierarchy	416
11.3	Files and Streams	418
11.4	Creating a Sequential-Access File	419
11.5	Reading Data from a Sequential-Access File	424
11.6	Random-Access Files	429
11.7	Creating a Randomly Accessed File	429
11.8	Writing Data Randomly to a Randomly Accessed File	431
11.9	Reading Data Randomly from a Randomly Accessed File	433
11.10	Case Study: A Transaction-Processing Program	435
<b>12</b>	<b>C Data Structures</b>	<b>450</b>
12.1	Introduction	451
12.2	Self-Referential Structures	452
12.3	Dynamic Memory Allocation	452
12.4	Linked Lists	454
12.5	Stacks	461
12.6	Queues	467
12.7	Trees	473
<b>13</b>	<b>C Preprocessor</b>	<b>502</b>
13.1	Introduction	503
13.2	The <code>#include</code> Preprocessor Directive	503
13.3	The <code>#define</code> Preprocessor Directive: Symbolic Constants	504
13.4	The <code>#define</code> Preprocessor Directive: Macros	504
13.5	Conditional Compilation	506
13.6	The <code>#error</code> and <code>#pragma</code> Preprocessor Directives	507

13.7	The # and ## Operators	507
13.8	Line Numbers	508
13.9	Predefined Symbolic Constants	508
13.10	Assertions	509
<b>14</b>	<b>Advanced C Topics</b>	<b>514</b>
14.1	Introduction	515
14.2	Redirecting Input/Output on UNIX and DOS Systems	515
14.3	Variable-Length Argument Lists	516
14.4	Using Command-Line Arguments	518
14.5	Notes on Compiling Multiple-Source-File Programs	520
14.6	Program Termination with exit and atexit	521
14.7	The <b>volatile</b> Type Qualifier	523
14.8	Suffixes for Integer and Floating-Point Constants	523
14.9	More on Files	523
14.10	Signal Handling	526
14.11	Dynamic Memory Allocation: Functions calloc and realloc	528
14.12	The Unconditional Branch: goto	529
<b>15</b>	<b>C++ as a “Better C”</b>	<b>536</b>
15.1	Introduction	537
15.2	C++	537
15.3	A Simple Program: Adding Two Integers	538
15.4	C++ Standard Library	540
15.5	Header Files	541
15.6	Inline Functions	543
15.7	References and Reference Parameters	545
15.8	Default Arguments and Empty Parameter Lists	549
15.9	Unary Scope Resolution Operator	551
15.10	Function Overloading	554
15.11	Function Templates	555
<b>16</b>	<b>C++ Classes and Data Abstraction</b>	<b>563</b>
16.1	Introduction	564
16.2	Implementing a Time Abstract Data Type with a Class	565
16.3	Class Scope and Accessing Class Members	572
16.4	Separating Interface from Implementation	573
16.5	Controlling Access to Members	577
16.6	Access Functions and Utility Functions	580
16.7	Initializing Class Objects: Constructors	583
16.8	Using Default Arguments with Constructors	584
16.9	Using Destructors	588
16.10	When Constructors and Destructors Are Called	588
16.11	Using Data Members and Member Functions	591
16.12	A Subtle Trap: Returning a Reference to a <b>private</b> Data Member	596
16.13	Assignment by Default Memberwise Copy	598
16.14	Software Reusability	600

<b>17</b>	<b>C++ Classes: Part II</b>	<b>610</b>
17.1	Introduction	611
17.2	<b>const</b> (Constant) Objects and <b>const</b> Member Functions	611
17.3	Composition: Objects as Members of Classes	620
17.4	<b>friend</b> Functions and <b>friend</b> Classes	625
17.5	Using the <b>this</b> Pointer	628
17.6	Dynamic Memory Allocation with Operators <b>new</b> and <b>delete</b>	633
17.7	<b>static</b> Class Members	635
17.8	Data Abstraction and Information Hiding	640
17.8.1	Example: Array Abstract Data Type	641
17.8.2	Example: String Abstract Data Type	642
17.8.3	Example: Queue Abstract Data Type	642
17.9	Container Classes and Iterators	643
<b>18</b>	<b>C++ Operator Overloading</b>	<b>650</b>
18.1	Introduction	651
18.2	Fundamentals of Operator Overloading	652
18.3	Restrictions on Operator Overloading	653
18.4	Operator Functions as Class Members vs. as friend Functions	655
18.5	Overloading Stream-Insertion and Stream-Extraction Operators	656
18.6	Overloading Unary Operators	659
18.7	Overloading Binary Operators	659
18.8	Case Study: An Array Class	660
18.9	Converting between Types	672
18.10	Overloading <b>++</b> and <b>--</b>	673
<b>19</b>	<b>C++ Inheritance</b>	<b>682</b>
19.1	Introduction	683
19.2	Inheritance: Base Classes and Derived Classes	685
19.3	Protected Members	687
19.4	Casting Base-Class Pointers to Derived-Class Pointers	687
19.5	Using Member Functions	693
19.6	Overriding Base-Class Members in a Derived Class	693
19.7	Public, Protected and Private Inheritance	698
19.8	Direct Base Classes and Indirect Base Classes	699
19.9	Using Constructors and Destructors in Derived Classes	699
19.10	Implicit Derived-Class Object to Base-Class Object Conversion	703
19.11	Software Engineering with Inheritance	704
19.12	Composition vs. Inheritance	705
19.13	“Uses A” and “Knows A” Relationships	706
19.14	Case Study: Point, Circle, Cylinder	706
<b>20</b>	<b>C++ Virtual Functions and Polymorphism</b>	<b>720</b>
20.1	Introduction	721
20.2	Type Fields and <b>switch</b> Statements	721
20.3	Virtual Functions	722

20.4	Abstract Base Classes and Concrete Classes	723
20.5	Polymorphism	723
20.6	New Classes and Dynamic Binding	725
20.7	Virtual Destructors	726
20.8	Case Study: Inheriting Interface and Implementation	726
20.9	Polymorphism, <b>virtual</b> Functions and Dynamic Binding “Under the Hood”	735
<b>21</b>	<b>C++ Stream Input/Output</b>	<b>743</b>
21.1	Introduction	745
21.2	Streams	745
21.2.1	Iostream Library Header Files	746
21.2.2	Stream Input/Output Classes and Objects	746
21.3	Stream Output	748
21.3.1	Stream-Insertion Operator	748
21.3.2	Cascading Stream-Insertion/Extraction Operators	750
21.3.3	Output of <b>char *</b> Variables	751
21.3.4	Character Output with Member Function <b>put</b> ; Cascading <b>puts</b>	752
21.4	Stream Input	752
21.4.1	Stream-Extraction Operator	752
21.4.2	<b>get</b> and <b>getline</b> Member Functions	755
21.4.3	<b>istream</b> Member Functions <b>peek</b> , <b>putback</b> and <b>ignore</b>	758
21.4.4	Type-Safe I/O	758
21.5	Unformatted I/O with read, <b>gcount</b> and write	758
21.6	Stream Manipulators	759
21.6.1	Integral Stream Base: <b>dec</b> , <b>oct</b> , <b>hex</b> and <b>setbase</b>	759
21.6.2	Floating-Point Precision ( <b>precision</b> , <b>setprecision</b> )	760
21.6.3	Field Width ( <b>setw</b> , <b>width</b> )	762
21.6.4	User-Defined Manipulators	763
21.7	Stream Format States	764
21.7.1	Format State Flags	764
21.7.2	Trailing Zeros and Decimal Points ( <b>ios::showpoint</b> )	765
21.7.3	Justification ( <b>ios::left</b> , <b>ios::right</b> , <b>ios::internal</b> )	766
21.7.4	Padding ( <b>fill</b> , <b>setfill</b> )	768
21.7.5	Integral Stream Base ( <b>ios::dec</b> , <b>ios::oct</b> , <b>ios::hex</b> , <b>ios::showbase</b> )	770
21.7.6	Floating-Point Numbers; Scientific Notation ( <b>ios::scientific</b> , <b>ios::fixed</b> )	771
21.7.7	Uppercase/Lowercase Control ( <b>ios::uppercase</b> )	772
21.7.8	Setting and Resetting the Format Flags ( <b>flags</b> , <b>setiosflags</b> , <b>resetiosflags</b> )	772
21.8	Stream Error States	774
21.9	Tying an Output Stream to an Input Stream	776
<b>22</b>	<b>C++ Templates</b>	<b>788</b>
22.1	Introduction	789
22.2	Class Templates	789
22.3	Class Templates and Nontype Parameters	795

22.4	Templates and Inheritance	796
22.5	Templates and friends	796
22.6	Templates and static Members	797
<b>23</b>	<b>C++ Exception Handling</b>	<b>801</b>
23.1	Introduction	802
23.2	When Exception Handling Should Be Used	804
23.3	Other Error-Handling Techniques	805
23.4	Basics of C++ Exception Handling: <b>try, throw, catch</b>	806
23.5	A Simple Exception-Handling Example: Divide by Zero	806
23.6	Throwing an Exception	809
23.7	Catching an Exception	810
23.8	Rethrowing an Exception	813
23.9	Exception Specifications	815
23.10	Processing Unexpected Exceptions	815
23.11	Stack Unwinding	816
23.12	Constructors, Destructors and Exception Handling	817
23.13	Exceptions and Inheritance	818
23.14	Processing <b>new</b> Failures	818
23.15	Class <b>auto_ptr</b> and Dynamic Memory Allocation	822
23.16	Standard Library Exception Hierarchy	824
<b>24</b>	<b>Introduction to Java Applications and Applets</b>	<b>834</b>
24.1	Introduction	835
24.2	Basics of a Typical Java Environment	836
24.3	General Notes about Java and This Book	839
24.4	A Simple Program: Printing a Line of Text	840
24.5	Another Java Application: Adding Integers	848
24.6	Sample Applets from the Java 2 Software Development Kit	853
	24.6.1 The TicTacToe Applet	854
	24.6.2 The DrawTest Applet	856
	24.6.3 The Java2D Applet	858
24.7	A Simple Java Applet: Drawing a String	859
24.8	Two More Simple Applets: Drawing Strings and Lines	866
24.9	Another Java Applet: Adding Integers	868
<b>25</b>	<b>Beyond C &amp; C++: Operators, Methods &amp; Arrays in Java</b>	<b>888</b>
25.1	Introduction	889
25.2	Primitive Data Types and Keywords	889
25.3	Logical Operators	891
25.4	Method Definitions	896
25.5	Java API Packages	901
25.6	Random Number Generation	906
25.7	Example: A Game of Chance	909
25.8	Methods of Class <b>JApplet</b>	917
25.9	Declaring and Allocating Arrays	919

25.10	Examples Using Arrays	920
25.11	References and Reference Parameters	929
25.12	Multiple-Subscripted Arrays	931
<b>26</b>	<b>Java Object-Based Programming</b>	<b>946</b>
26.1	Introduction	947
26.2	Implementing a Time Abstract Data Type with a Class	948
26.3	Class Scope	956
26.4	Creating Packages	956
26.5	Initializing Class Objects: Constructors	960
26.6	Using <i>Set</i> and <i>Get</i> Methods	961
26.7	Using the <b>this</b> Reference	968
26.8	Finalizers	971
26.9	Static Class Members	971
<b>27</b>	<b>Java Object-Oriented Programming</b>	<b>984</b>
27.1	Introduction	985
27.2	Superclasses and Subclasses	987
27.3	<b>protected</b> Members	989
27.4	Relationship between Superclass Objects and Subclass Objects	989
27.5	Implicit Subclass-Object-to-Superclass-Object Conversion	997
27.6	Software Engineering with Inheritance	997
27.7	Composition vs. Inheritance	999
27.8	Introduction to Polymorphism	999
27.9	Type Fields and <b>switch</b> Statements	999
27.10	Dynamic Method Binding	999
27.11	<b>final</b> Methods and Classes	1000
27.12	Abstract Superclasses and Concrete Classes	1000
27.13	Polymorphism Example	1001
27.14	New Classes and Dynamic Binding	1003
27.15	Case Study: Inheriting Interface and Implementation	1004
27.16	Case Study: Creating and Using Interfaces	1010
27.17	Inner Class Definitions	1015
27.18	Notes on Inner Class Definitions	1026
27.19	Type-Wrapper Classes for Primitive Types	1026
<b>28</b>	<b>Java Graphics and Java2D</b>	<b>1037</b>
28.1	Introduction	1038
28.2	Graphics Contexts and Graphics Objects	1040
28.3	Color Control	1041
28.4	Font Control	1048
28.5	Drawing Lines, Rectangles and Ovals	1054
28.6	Drawing Arcs	1058
28.7	Drawing Polygons and Polylines	1060
28.8	The Java2D API	1063
28.9	Java2D Shapes	1063

<b>29</b>	<b>Java Graphical User Interface Components</b>	<b>1078</b>
29.1	Introduction	1079
29.2	Swing Overview	1081
29.3	<b>JLabel</b>	1083
29.4	Event Handling Model	1086
29.5	<b>JTextField</b> and <b>JPasswordField</b>	1088
29.5.1	How Event Handling Works	1093
29.6	<b>JTextArea</b>	1094
29.7	<b>JButton</b>	1097
29.8	<b>JCheckBox</b>	1101
29.9	<b>JComboBox</b>	1104
29.10	Mouse Event Handling	1106
29.11	Layout Managers	1111
29.11.1	<b>FlowLayout</b>	1111
29.11.2	<b>BorderLayout</b>	1114
29.11.3	<b>GridLayout</b>	1117
29.12	Panels	1119
29.13	Creating a Self-Contained Subclass of <b>JPanel</b>	1121
29.14	Windows	1127
29.15	Using Menus with Frames	1129
<b>30</b>	<b>Java Multimedia: Images, Animation, Audio and Video</b>	<b>1150</b>
30.1	Introduction	1151
30.2	Loading, Displaying and Scaling Images	1152
30.3	Loading and Playing Audio Clips	1155
30.4	Animating a Series of Images	1158
30.5	Animation Issues	1162
30.6	Customizing Applets via the HTML <b>param</b> Tag	1164
30.7	Image Maps	1169
30.8	Java Plug-In	1171
30.9	Internet and World Wide Web Resources	1174
<b>A</b>	<b>Internet and Web Resources</b>	<b>1180</b>
A.1	C/C++ Resources	1180
A.2	C++ Tutorials	1181
A.3	C/C++ FAQs	1182
A.4	Visual C++	1182
A.5	comp.lang.c++	1182
A.6	C/C++ Compilers	1184
A.7	C++ Development Tools	1185
A.8	Java Resources	1185
A.9	Java Products	1187
A.10	Java FAQs	1187
A.11	Java Tutorials	1187
A.12	Java Magazines	1188

A.13	Java Applets	1188
A.14	Multimedia	1189
A.15	Java Newsgroups	1189
<b>B</b>	<b>C99 Internet and Web Resources</b>	<b>1191</b>
B.1	C99 Resources	1191
<b>C</b>	<b>Operator Precedence Charts</b>	<b>1193</b>
<b>D</b>	<b>ASCII Character Set</b>	<b>1198</b>
<b>E</b>	<b>Number Systems</b>	<b>1199</b>
E.1	Introduction	1200
E.2	Abbreviating Binary Numbers as Octal Numbers and Hexadecimal Numbers	1203
E.3	Converting Octal Numbers and Hexadecimal Numbers to Binary Numbers	1204
E.4	Converting from Binary, Octal, or Hexadecimal to Decimal	1204
E.5	Converting from Decimal to Binary, Octal, or Hexadecimal	1205
E.6	Negative Binary Numbers: Two's Complement Notation	1206
	<b>Index</b>	<b>1212</b>