

# Contents

<b>Foreword</b> .....	<b>vii</b>
<b>Preface</b> .....	<b>ix</b>
<b>1 Object-Oriented Paradigm</b> .....	<b>1</b>
1.1 Why New Programming Paradigms ? .....	1
1.2 OOPs ! a New Paradigm .....	2
1.3 Evolution of Programming Paradigms .....	5
1.4 Structured Versus Object-Oriented Development .....	9
1.5 Elements of Object-Oriented Programming .....	11
1.6 Objects .....	12
1.7 Classes .....	13
1.8 Multiple Views of the Same Object .....	15
1.9 Encapsulation and Data Abstraction .....	16
1.10 Inheritance .....	17
1.11 Delegation - Object Composition .....	19
1.12 Polymorphism .....	20
1.13 Message Communication .....	20
1.14 Popular OOP Languages .....	21
1.15 Merits and Demerits of OO Methodology .....	25
1.16 OO Learning Curve .....	26
1.17 Software Reuse .....	27
1.18 Objects Hold the Key .....	30
Review Questions .....	31
<b>2 Moving from C to C++</b> .....	<b>32</b>
2.1 Introduction .....	32
2.2 Hello World .....	32
2.3 Streams Based I/O .....	35
2.4 Single Line Comment .....	40
2.5 Literals–Constant Qualifiers .....	42
2.6 Scope Resolution Operator :: .....	44
2.7 Variable Definition at the Point of Use .....	45
2.8 Variable Aliases–Reference Variables .....	47
2.9 Strict Type Checking .....	49
2.10 Parameters Passing by Reference .....	51
2.11 Inline Functions .....	53
2.12 Function Overloading .....	54
2.13 Default Arguments .....	57
2.14 Keyword typedef .....	59
2.15 Functions as a Part of a Struct .....	60
2.16 Type Conversion .....	62
2.17 Function Templates .....	64
2.18 Runtime Memory Management .....	67
Review Questions .....	71

<b>3 C++ at a Glance .....</b>	<b>73</b>
3.1 Introduction .....	73
3.2 Data Encapsulation and Abstraction—Classes .....	73
3.3 Inheritance—Derived Classes .....	78
3.4 Polymorphism—Operator Overloading .....	81
3.5 Friend Functions .....	84
3.6 Polymorphism—Virtual Functions .....	87
3.7 Generic Classes—Class Templates .....	90
3.8 Exception Handling .....	92
3.9 Streams Computation .....	95
Review Questions .....	99
<b>4 Data Types, Operators and Expressions .....</b>	<b>100</b>
4.1 Introduction .....	100
4.2 Character Set .....	100
4.3 Tokens, Identifiers, and Keywords .....	101
4.4 Variables .....	102
4.5 Data Types and Sizes .....	103
4.6 Variable Definition .....	104
4.7 Variable Initialization .....	104
4.8 Characters and Character Strings .....	106
4.9 Operators and Expressions .....	107
4.10 Qualifiers .....	109
4.11 Arithmetic Operators .....	112
4.12 Relational Operators .....	115
4.13 Logical Operators .....	118
4.14 Bit-wise Operators .....	120
4.15 Compound Assignment Operators .....	124
4.16 Increment and Decrement Operators .....	125
4.17 Conditional Operator (Ternary Operator) .....	126
4.18 Special Operators .....	128
4.19 typedef Statement .....	128
4.20 Promotion and Type Conversion .....	129
4.21 Constants .....	131
4.22 Declaring Symbolic Constants—Literals .....	134
4.23 Enumerated Data Types .....	136
4.24 Macro Functions .....	138
4.25 Operator Precedence and Associativity .....	140
Review Questions .....	142
<b>5 Control Flow .....</b>	<b>143</b>
5.1 Introduction .....	143
5.2 Statements and Block .....	143
5.3 if Statement .....	144
5.4 if-else Statement .....	146
5.5 Nested if-else Statements .....	148

5.6 for Loop .....	149
5.7 while loop .....	154
5.8 do..while Loop .....	156
5.9 break Statement .....	157
5.10 switch Statement .....	160
5.11 continue Statement .....	162
5.12 goto Statement .....	164
5.13 Wild Statements .....	165
Review Questions .....	167
<b>6 Arrays and Strings .....</b>	<b>168</b>
6.1 Introduction .....	168
6.2 Operations on Arrays .....	168
6.3 Array Illustrations .....	174
6.4 Multi-dimensional Arrays .....	178
6.5 Strings .....	182
6.6 Strings Manipulations .....	184
6.7 Arrays of Strings .....	187
6.8 Evaluation Order / Undefined Behaviors .....	188
Review Questions .....	189
<b>7 Modular Programming with Functions .....</b>	<b>191</b>
7.1 Introduction .....	191
7.2 Function Components .....	193
7.3 Passing Data to Functions .....	198
7.4 Function Return Data Type .....	201
7.5 Library Functions .....	203
7.6 Parameter Passing .....	204
7.7 Return by Reference .....	209
7.8 Default Arguments .....	210
7.9 Inline Functions .....	213
7.10 Function Overloading .....	214
7.11 Function Templates .....	219
7.12 Arrays and Functions .....	220
7.13 C++ Stack .....	221
7.14 Scope and Extent of Variables .....	223
7.15 Storage Classes .....	225
7.16 Functions with Variable Number of Arguments .....	228
7.17 Recursive Functions .....	231
7.18 Complete Syntax of main() .....	234
Review Questions .....	236
<b>8 Structures and Unions .....</b>	<b>237</b>
8.1 Introduction .....	237
8.2 Structure Declaration .....	237
8.3 Structure Definition .....	238

8.4	Accessing Structure Members .....	240
8.5	Structure Initialization .....	242
8.6	Nesting of Structures .....	243
8.7	Array of Structures .....	246
8.8	Structures and Functions .....	251
8.9	Data Type Enhancement Using typedef .....	255
8.10	Structures and Encapsulation .....	257
8.11	Unions .....	259
8.12	Differences between Structures and Unions .....	261
8.13	Bit-fields in Structures .....	264
	Review Questions .....	267
<b>9</b>	<b>Pointers and Runtime Binding .....</b>	<b>268</b>
9.1	Introduction .....	268
9.2	Pointers and their Binding .....	269
9.3	Address Operator & .....	270
9.4	Pointer Variables .....	271
9.5	Void Pointers .....	276
9.6	Pointer Arithmetic .....	278
9.7	Runtime Memory Management .....	282
9.8	Pointers to Pointers .....	283
9.9	Array of Pointers .....	285
9.10	Dynamic Multi-dimensional Arrays .....	289
9.11	Pointer Constants .....	294
9.12	Pointers and String Functions .....	294
9.13	Environment Specific Issues .....	296
9.14	Pointers to Functions .....	298
9.16	Pointers to Constant Objects .....	301
9.17	Constant Pointers .....	302
9.18	Pointer to Structures .....	302
9.19	Wild Pointers .....	307
	Review Questions .....	310
<b>10</b>	<b>Classes and Objects .....</b>	<b>313</b>
10.1	Introduction .....	313
10.2	Class Specification .....	314
10.3	Class Objects .....	316
10.4	Accessing Class Members .....	317
10.5	Defining Member Functions .....	321
10.6	Outside Member Functions as inline .....	325
10.7	Accessing Member Functions within the Class .....	328
10.8	Data Hiding .....	329
10.9	Access Boundary of Objects Revisited .....	332
10.10	Empty Classes .....	334
10.11	Pointers within a Class .....	334
10.12	Passing Objects as Arguments .....	336

10.13	Returning Objects from Functions .....	340
10.14	Friend Functions and Friend Classes .....	342
10.15	Constant Parameters and Member Functions .....	349
10.16	Structures and Classes .....	352
10.17	Static Data and Member Functions .....	354
10.18	Class, Objects and Memory Resource .....	358
10.19	Class Design Steps .....	360
	Review Questions .....	361
<b>11</b>	<b>Object Initialization and Cleanup .....</b>	<b>363</b>
11.1	Class Revisited .....	363
11.2	Constructors .....	364
11.3	Parameterized Constructors .....	368
11.4	Destructor .....	371
11.5	Constructor Overloading .....	373
11.6	Order of Construction and Destruction .....	376
11.7	Constructors with Default Arguments .....	377
11.8	Nameless Objects .....	380
11.9	Dynamic Initialization through Constructors .....	381
11.10	Constructors with Dynamic Operations .....	383
11.11	Copy Constructor .....	385
11.12	Constructors for Two-dimensional Arrays .....	387
11.13	Constant Objects and Constructor .....	393
11.14	Static Data Members with Constructors and Destructors .....	395
11.15	Nested Classes .....	397
	Review Questions .....	398
<b>12</b>	<b>Dynamic Objects .....</b>	<b>400</b>
12.1	Introduction .....	400
12.2	Pointers to Objects .....	400
12.3	Live Objects .....	408
12.4	Array of Objects .....	411
12.5	Array of Pointers to Objects .....	413
12.6	Pointers to Object Members .....	416
12.7	Function set_new_handler() .....	421
12.8	this Pointer .....	422
12.9	Self-referential Classes .....	424
12.10	Guidelines for Passing Object Parameters .....	431
	Review Questions .....	431
<b>13</b>	<b>Operator Overloading .....</b>	<b>432</b>
13.1	Introduction .....	432
13.2	Overloadable Operators .....	433
13.3	Unary Operator Overloading .....	434
13.4	operator Keyword .....	436
13.5	Operator Return Values .....	438

13.6	Nameless Temporary Objects .....	439
13.7	Limitations of Increment/Decrement Operators .....	441
13.8	Binary Operator Overloading .....	445
13.9	Arithmetic Operators .....	446
13.10	Concatenation of Strings .....	452
13.11	Comparison Operators .....	454
13.12	Arithmetic Assignment Operators .....	457
13.13	Overloading of new and delete Operators .....	462
13.14	Data Conversion .....	464
13.15	Conversion Between Basic Data Types .....	464
13.16	Conversion Between Objects and Basic Types .....	465
13.17	Conversion between Objects of Different Classes .....	470
13.18	Subscript Operator Overloading .....	477
13.19	Overloading with Friend Functions .....	480
13.20	Assignment Operator Overloading .....	488
13.21	Tracing Memory Leaks .....	491
13.22	Niceties of Operator Overloading and Conversions .....	493
	Review Questions .....	497
<b>14</b>	<b>Inheritance .....</b>	<b>499</b>
14.1	Introduction .....	499
14.2	Class Revisited .....	500
14.3	Derived Class Declaration .....	503
14.4	Forms of Inheritance .....	510
14.5	Inheritance and Member Accessibility .....	511
14.6	Constructors in Derived Classes .....	516
14.7	Destructors in Derived Classes .....	523
14.8	Constructors Invocation and Data Members Initialization .....	525
14.9	Overloaded Member Functions .....	528
14.10	Abstract Classes .....	533
14.11	Multilevel Inheritance .....	533
14.12	Multiple Inheritance .....	537
14.13	Hierarchical Inheritance .....	548
14.14	Multipath Inheritance and Virtual Base Classes .....	552
14.15	Hybrid Inheritance .....	558
14.16	Object Composition–Delegation .....	562
14.17	When to Use Inheritance ? .....	567
14.18	Benefits of inheritance .....	567
14.19	Cost of Inheritance .....	568
	Review Questions .....	568
<b>15</b>	<b>Virtual Functions .....</b>	<b>570</b>
15.1	Introduction .....	570
15.2	Need for Virtual Functions .....	571
15.3	Pointer to Derived Class Objects .....	574
15.4	Definition of Virtual Functions .....	578

15.5	Array of Pointers to Base Class Objects .....	581
15.6	Pure Virtual Functions .....	584
15.7	Abstract Classes .....	585
15.8	Virtual Destructors .....	588
15.9	How is Dynamic Binding Achieved ? .....	591
15.10	Rules for Virtual Functions .....	593
	Review Questions .....	594
<b>16</b>	<b>Generic Programming with Templates .....</b>	<b>596</b>
16.1	Introduction .....	596
16.2	Function Templates .....	596
16.3	Overloaded Function Templates .....	605
16.4	Nesting of Function Calls .....	606
16.5	Multiple Arguments Function Template .....	607
16.6	User Defined Template Arguments .....	609
16.7	Class Templates .....	610
16.8	Inheritance of Class Template .....	616
16.9	Class Template Containership .....	620
16.10	Class Template with Overloaded Operators .....	626
	Review Questions .....	627
<b>17</b>	<b>Streams Computation with Console .....</b>	<b>629</b>
17.1	What are Streams ? .....	629
17.2	Predefined Console Streams .....	631
17.3	Hierarchy of Console Stream Classes .....	634
17.4	Unformatted I/O Operations .....	635
17.5	Formatted Console I/O Operations .....	640
17.6	Manipulators .....	648
17.7	Custom/User-Defined Manipulators .....	655
17.8	Stream Operator with User-defined Classes .....	660
	Review Questions .....	662
<b>18</b>	<b>Streams Computation with Files .....</b>	<b>664</b>
18.1	Introduction .....	664
18.2	Hierarchy of File Stream Classes .....	665
18.3	Opening and Closing of Files .....	667
18.4	Testing for Errors .....	672
18.5	File Modes .....	677
18.6	File Pointers and their Manipulations .....	678
18.7	Sequential Access to a File .....	682
18.8	ASCII and Binary Files .....	683
18.9	Saving and Retrieving of Objects .....	685
18.10	File Input/Output with fstream Class .....	689
18.11	Random Access to a File .....	691
18.12	In-Memory Buffers and Data Formatting .....	695
18.13	Error Handling During File Manipulations .....	696

18.14	Filter Utilities .....	699
	Review Questions .....	702
<b>19</b>	<b>Exception Handling .....</b>	<b>703</b>
19.1	Introduction .....	703
19.2	Error Handling .....	703
19.3	Exception Handling Model .....	704
19.4	Exception Handling Constructs .....	705
19.5	Handler Throwing the Same Exception Again .....	711
19.6	List of Exceptions .....	713
19.7	Catch All Exceptions .....	717
19.8	Exceptions in Constructors and Destructors .....	719
19.9	Handling Uncaught Exceptions .....	721
19.10	Exceptions in Operator Overloaded Functions .....	727
19.11	Exceptions in Inheritance Tree .....	729
19.12	Exceptions in Class Templates .....	731
19.13	Fault Tolerant Design Techniques .....	735
19.14	Case-Study on Software Fault Tolerance .....	737
19.15	Memory Allocation Failure Exception .....	740
19.16	Ten Rules for Handling Exceptions Successfully .....	742
	Review Questions .....	747
<b>20</b>	<b>OO Analysis, Design and Development .....</b>	<b>748</b>
20.1	Software Life Cycle: Water-Fall Model .....	749
20.2	Cost of Error Correction .....	751
20.3	Change Management .....	752
20.4	Reusable Components .....	753
20.5	Software Life Cycle: Fountain-Flow Model .....	755
20.6	Object-Oriented Notations .....	756
20.7	Object-Oriented Methodologies .....	756
20.8	Coad and Yourdon Object-Oriented Analysis .....	759
20.9	Booch's Object-Oriented Design .....	760
20.10	Class Design .....	761
20.11	How to Build Reliable Code ? .....	764
20.12	OO Software Performance Tuning .....	765
20.13	Software Project Management .....	766
20.14	Plan for OO Battle .....	768
20.15	A Final Word .....	769
	Review Questions .....	769
<b>Appendices</b>		
A.	C++ Keywords and Operators .....	770
B.	C++ Library Functions .....	788
C.	Glossary .....	790
D.	ASCII Character Set .....	794
E.	Bibliography .....	798
F.	Index .....	799