

Eastern  
Economy  
Edition

*Second Edition*

# C++ and Object-Oriented Programming Paradigm



Debasish Jana

# Contents

<i>Preface</i>		<i>xi</i>
<i>Acknowledgements</i>		<i>xv</i>
<b>1. OVERVIEW</b>		<b>1-33</b>
<i>Learning Objectives</i>	1	
1.1 Introduction	1	
1.1.1 Basics of Programming	2	
1.1.2 Language Translators	3	
1.1.3 Programming Paradigms	5	
1.2 Need for Object-Oriented Programming	9	
1.3 Basics of OOP	10	
1.4 OO Languages	10	
1.5 Evolution of C++	12	
1.5.1 Structure of a C++ Program	13	
1.5.2 Some Terminologies	14	
1.6 First C++ Program	19	
1.6.1 Input and Output	21	
1.6.2 Compilation	23	
1.7 Getting Familiar with the OOP Terms	27	
1.7.1 Class and Object	27	
1.7.2 Abstraction and Encapsulation	28	
1.7.3 Polymorphism	29	
1.7.4 Inheritance	29	
1.7.5 Static and Dynamic Binding	30	
<i>Summary</i>	31	
<i>Review Questions</i>	33	
<b>2. DECLARATIONS AND EXPRESSIONS</b>		<b>34-58</b>
<i>Learning Objectives</i>	34	
2.1 Introduction	34	
2.1.1 Fundamental Data Types	35	
2.1.2 Qualifiers to Data Types	37	
2.1.3 Reference Data Types	39	

- 2.1.4 Variables 39
- 2.1.5 Constants 41
- 2.1.6 Operators and Expressions 43
- 2.1.7 Operator Precedence and Associativity 54

*Summary* 55

*Review Questions* 56

*Annexure* 57

### 3. STATEMENTS

59-89

*Learning Objectives* 59

- 3.1 Introduction 59
  - 3.1.1 Labeled Statement 60
  - 3.1.2 Expression Statement 60
  - 3.1.3 Compound Statement 61
  - 3.1.4 Control Statement 62
  - 3.1.5 Jump Statement 84
  - 3.1.6 Declaration Statement 87
  - 3.1.7 Try-Throw-Catch Statements 87

*Summary* 87

*Review Questions* 88

### 4. ARRAY, POINTER AND STRUCTURE

90-124

*Learning Objectives* 90

- 4.1 Introduction 90
  - 4.1.1 Array 91
  - 4.1.2 Addresses and Pointers 103
  - 4.1.3 Pointers and Functions 116

*Summary* 122

*Review Questions* 123

### 5. FUNCTIONS

125-151

*Learning Objectives* 125

- 5.1 Introduction 125
  - 5.1.1 Declaration, Definition and Call 127
  - 5.1.2 Inline Functions 131
  - 5.1.3 main Function Arguments 132
  - 5.1.4 Reference Variables 133
  - 5.1.5 Function Overloading 135
  - 5.1.6 Default Arguments 136
  - 5.1.7 Parameter Passing 138
  - 5.1.8 Recursion 141
  - 5.1.9 Scope of Variables 144
  - 5.1.10 Return-by-value and Return-by-reference 146
  - 5.1.11 Pointer to Functions 147

*Summary* 148

*Review Questions* 149

- 6. PREPROCESSOR DIRECTIVES** 152-165
- Learning Objectives* 152
- 6.1 Introduction 152
- 6.1.1 Phases of Preprocessing 154
  - 6.1.2 Trigraph Sequences 154
  - 6.1.3 Digraph Characters 155
  - 6.1.4 #define 155
  - 6.1.5 The # Operator 156
  - 6.1.6 The Null Directive 157
  - 6.1.7 The ## Operator 158
  - 6.1.8 #undef 159
  - 6.1.9 #ifdef, #ifndef, #if, #endif, #else and #elif 159
  - 6.1.10 #line 160
  - 6.1.11 #error 163
  - 6.1.12 #include 163
  - 6.1.13 #pragma 164
- Summary* 164
- Review Questions* 165
- 7. STANDARD C LIBRARY FUNCTIONS AND STANDARD HEADER FILES** 166-189
- Learning Objectives* 166
- 7.1 Introduction 166
- 7.1.1 Why Library? 166
  - 7.1.2 Header Files 167
- Summary* 188
- Review Questions* 189
- 8. DATA ABSTRACTION THROUGH CLASSES AND USER-DEFINED DATA TYPES** 190-229
- Learning Objectives* 190
- 8.1 Introduction 190
- 8.1.1 C-Structure 191
  - 8.1.2 typedef 193
- 8.2 Class 195
- 8.2.1 Class Members 200
  - 8.2.2 Controlling Access to Members of a Class 202
  - 8.2.3 Constructor and Destructor 203
  - 8.2.4 Copy Constructor 210
- 8.3 Dynamic Memory Management 213
- 8.3.1 Operators new and delete 214
  - 8.3.2 malloc and free 219
  - 8.3.3 Static Member 224
  - 8.3.4 Scope of Class Names 224
  - 8.3.5 Scope of Variables 225
- Summary* 227
- Review Questions* 228

- 9. OPERATOR OVERLOADING** 230-259
- Learning Objectives* 230
- 9.1 Introduction 230
- 9.1.1 Restrictions 239
- 9.1.2 Overloading Unary Operators 240
- 9.1.3 Overloading Binary Operators 241
- 9.1.4 Overloaded Function Calls 242
- 9.1.5 Overloaded Subscripting 243
- 9.1.6 Overloaded Class Member Access 244
- 9.1.7 Cast Operator 245
- 9.1.8 User-defined Conversions 246
- 9.1.9 Overloaded Increment and Decrement 247
- 9.1.10 Overloaded Non-member Operator 251
- 9.1.11 Overloaded *new* and *delete* 253
- Summary* 256
- Review Questions* 257
- 10. CLASS RELATIONSHIPS** 260-302
- Learning Objectives* 260
- 10.1 Introduction 260
- 10.1.1 Characteristics of OOP 261
- 10.1.2 Relationships 261
- 10.2 Polymorphism 262
- 10.2.1 Coercion 263
- 10.2.2 Overloading 264
- 10.2.3 Parametric Polymorphism 265
- 10.2.4 Inclusion Polymorphism 265
- 10.3 Inheritance 266
- 10.3.1 Direct and Indirect Superclasses 268
- 10.3.2 Multiple Inheritance 274
- 10.3.3 Virtual Base Classes 276
- 10.3.4 Friend 282
- 10.3.5 Per Class Protection 289
- 10.3.6 Virtual Function 289
- 10.3.7 Abstract Class 291
- 10.3.8 Overriding and Hiding 292
- 10.3.9 Dynamic Binding of Functions 293
- 10.3.10 Virtual Destructor 296
- 10.3.11 Virtual Operators 297
- 10.3.12 Accessibility in Derived Classes 297
- 10.3.13 Linking C file in C++ program 299
- Summary* 299
- Review Questions* 300
- 11. ADVANCED CONCEPTS** 303-347
- Learning Objectives* 303
- 11.1 Introduction 303

11.2	Template	304	
✓ 11.2.1	Class Template	304	
11.2.2	Member Function Inclusion	308	
✓ 11.2.3	Function Template	309	
11.2.4	Parameter Values for Templates	314	
11.2.5	Template Specialization	320	
11.2.6	Template Inheritance	321	
11.2.7	Namespace	325	
11.2.8	Named Namespace	325	
11.2.9	Using Named Namespace	326	
11.2.10	Namespace Alias	327	
11.2.11	Unnamed Namespace	327	
✓ 11.3	Exception Handling	328	
11.3.1	Capturing Matching Typed Exception through Overloaded <i>Catch</i> Blocks	332	
11.3.2	Ellipsis in <i>Catch</i> Block	335	
11.3.3	Nested <i>Try-Catch</i> Blocks	335	
11.3.4	Rethrowing an Exception	336	
11.3.5	Conditional Expression in a <i>Throw</i> Expression	336	
11.3.6	Constructors and Destructors in Exception Handling	336	
11.3.7	Run-time Standard Exceptions	336	
11.4	Advanced Casting Operators	337	
11.4.1	<code>static_cast</code> Operator	339	
11.4.2	<code>dynamic_cast</code> Operator	341	
11.4.3	<code>reinterpret_cast</code> Operator	343	
11.4.4	<code>const_cast</code> Operator	344	
11.4.5	<code>typeid</code> Operator	344	
	<i>Summary</i>	345	
	<i>Review Questions</i>	346	
<b>12.</b>	<b>THE STANDARD LIBRARY IN C++</b>		<b>348-377</b>
	<i>Learning Objectives</i>	348	
12.1	Introduction	348	
12.2	Standard Library Functions	349	
12.2.1	Input and Output	349	
12.2.2	<code>iostream</code> Class Hierarchy	354	
12.2.3	Class <code>ios</code>	354	
12.2.4	Other Stream Classes	361	
12.2.5	Standard Template Library	369	
	<i>Summary</i>	376	
	<i>Review Questions</i>	377	
<b>13.</b>	<b>DATA STRUCTURES AND APPLICATIONS IN C++</b>		<b>378-408</b>
	<i>Learning Objectives</i>	378	
13.1	Introduction	378	
13.2	Array	383	
13.2.1	Searching	386	
13.2.2	Sorting	390	

13.3	Linked Lists	395	
13.4	A Small Example Program	403	
	<i>Summary</i>	407	
	<i>Review Questions</i>	408	
<b>14.</b>	<b>OBJECT-ORIENTED DESIGN AND MODELING</b>		<b>409–442</b>
	<i>Learning Objectives</i>	409	
14.1	Introduction	409	
14.2	Software Development	411	
14.3	Software Engineering Perspective	412	
14.3.1	The Desirable Qualities of Software Systems	412	
14.3.2	Software Architecture	414	
14.3.3	Software Process Life Cycle	418	
14.3.4	Object-Oriented Process	422	
14.3.5	Best Practices of Software Development	424	
14.3.6	Phases of Software Development—Inception, Elaboration, Construction, Transition	426	
14.3.7	Object-Oriented Principles and Concepts Revisited	429	
14.3.8	Classes and Objects	429	
14.3.9	Modularity	429	
14.3.10	Abstraction and Encapsulation	430	
14.3.11	Association, Aggregation and Composition	432	
14.3.12	Inheritance	432	
14.4	OO Methodology	433	
14.4.1	Need for Modeling	433	
14.4.2	Views of Booch, Jacobson and Rumbaugh Prior to UML	434	
14.4.3	UML Overview and History	435	
14.5	Object-Oriented Design Patterns	436	
	<i>Summary</i>	441	
	<i>Review Questions</i>	442	
<b>15.</b>	<b>UNIFIED MODELING LANGUAGE</b>		<b>443–483</b>
	<i>Learning Objectives</i>	443	
15.1	Introduction	443	
15.1.1	UML Building Blocks	446	
15.1.2	Use Case, Actors and Use Case Diagrams	450	
15.1.3	Structural Modeling	456	
15.1.4	Behavioral Modeling	469	
15.1.5	Packaging and Deployment	475	
15.1.6	UML and Software Development Process	480	
	<i>Summary</i>	482	
	<i>Review Questions</i>	483	
	<b>Problems (for Laboratory Workouts)</b>		<b>485–502</b>
	<b>Glossary</b>		<b>503–514</b>
	<b>Bibliography</b>		<b>515–516</b>
	<b>Index</b>		<b>517–531</b>