# Contents

## Preface  
**xi**

## Acknowledgements  
**xiii**

## PART-I Structured Programming

### 1. Introduction  
1.1 Overview  
**3**

1.2 Computer System Architecture  
**8**

1.3 C/C++ Development Environment  
**13**

1.4 Evolution of Programming Languages  
**17**

### 2. Fundamentals  
2.1 Overview  
**23**

2.2 The First C/C++ Program  
**23**

2.3 Writing Comments  
**27**

2.4 Constants or Literals  
**28**

2.5 Variables and Data Types  
**31**

2.6 `printf()` Statement (Supported by both C and C++)  
**38**

2.7 C++ Style of Printing the Value on Computer Screen  
**49**

2.8 `endl` Modifier (Supported by C++ not by C)  
**51**

2.9 Accepting User Input Using `scanf()` Function (Supported by both C and C++)  
**52**

2.10 `cin` Object in C++ (Only in C++ not in C)  
**58**

2.11 Manipulator `setw`  
**63**

2.12 Defining Constants using `#define`-A Pre-processor Directive  
**65**

2.13 Character Specific Input/Output  
**66**

### 3. Operators and Type Casting  
3.1 Overview  
**71**

3.2 Arithmetic Operators  
**77**

3.3 Relational Operators  
**82**
## Table of Contents

3.4 Shorthand Operators 84
3.5 Bitwise Operators in C/C++ 85
3.6 Increment/Decrement Operators 89
3.7 Order of Operations Evaluated by the `printf()` Statement 94
3.8 Implicit Type Casting/System Casting 97
3.9 Explicit Type Casting 97
3.10 `sizeof` Operator in C/C++ 101
3.11 Scope Resolution Operator (::) | Only in C++ not in C 102

4. **Decision Making Control Statements** 110
   4.1 Overview 110
   4.2 `if else` Statement 110
   4.3 Logical Operators 123
   4.4 `else if` Ladder 125
   4.5 `switch` Statement 132
   4.6 Ternary Operator/Conditional Operator 135
   4.7 `goto` Statement 139

5. **Iterative Control Statements: Loops** 146
   5.1 Introduction 146
   5.2 `while` Loop in C/C++ 149
   5.3 `for` Loop in C/C++ 164
   5.4 `do...while` Loop 180
   5.5 `break` and `continue` Statements 183
   5.6 Infinite Loops 190
   5.7 Comma Operator with `for` Loop 192
   5.8 Creating Variables Local to Loops (Possible in C++ but not in C) 192
   5.9 Empty Loops 193

6. **Arrays** 216
   6.1 Overview 216
   6.2 Creating an Array 218
   6.3 Array of Characters 238
   6.4 2D Arrays 252
   6.5 2D Array of Characters 261
   6.6 String-Specific Input and Output Operations: `gets()`/`puts()` 264

7. **Functions** 299
   7.1 Overview 299
   7.2 Creating Functions 303
   7.3 Local Variables of the Function 309
   7.4 Functions with Arguments 313
   7.5 Functions with Return Values 318
## Table of Contents

### 7.6 Passing Array as an Argument to the Function

7.7 Recursion

7.8 Activation frames: How Function Calls and Returns are Internally Handled in C/C++

7.9 Storage Classes in C/C++

7.10 Inline Functions in C/C++

7.11 Function with Default Arguments (Only in C++ not in C)

7.12 Command Line Arguments

7.13 Some Built-in Functions

### 8. Pointers

8.1 Overview

8.2 Creating Pointers

8.3 Data Type of Pointers

8.4 Types of Function Calls

8.5 Arithmetic Operations with Pointers

8.6 Accessing Array Elements using a Pointer

8.7 Initialization of an Array: Revisited

8.8 Self-addressability of Character Variables

8.9 Array of Pointers

8.10 Pointer to a Pointer

8.11 Pointers and 2D Arrays

8.12 `void` Pointers

8.13 Pointer to a Function

8.14 Reference Variables (Only Available in C++ not in C)

8.15 `Lvalue` and `Rvalue`

### 9. Structures and Unions

9.1 Overview

9.2 Creating Structures

9.3 Array of Structure Objects

9.4 Nesting of Structures

9.5 Structures and Pointers

9.6 Accessing Array of Objects using a Pointer

9.7 Passing Object as an Argument to a Function

9.8 Difference between Structure and Union

### 10. Dynamic Memory Allocation in C++

10.1 Overview

10.2 Dynamic Memory Management in C++

10.3 Linked List

10.4 `delete` Keyword in C++
PART-II Object Oriented Programming

11. Classes and Objects 533
   11.1 Overview 533
   11.2 Creating Classes 537
   11.3 Creating Objects of a Class 538
   11.4 Access Specifiers in C++ 542
   11.5 Data Hiding and Encapsulation 545
   11.6 Employee Management System: An Example 548
   11.7 Account Management System: An Example 554
   11.8 Calculating Slope of the Line: An Example 559
   11.9 Addition of Complex Numbers: An Example 563
   11.10 Addition of Points in Cartesian Coordinate System: An Example 567
   11.11 Array of Objects 569
   11.12 Employee Management System: Revisited 571
   11.13 friend Functions 573
   11.14 Addition of Point Objects using friend Function: An Example 574
   11.15 Pointer to Objects 580
   11.16 Binding of Pointers with Individual Members of the Class 584
   11.17 this Pointer 589
   11.18 Resolving Ambiguity using this Pointer 592
   11.19 Cloning Objects using this Pointer: An Example 594
   11.20 Dynamic Memory Allocation of Objects 596
   11.21 Linked List to Maintain Data about Employees 600
   11.22 Composition and Aggregation between Classes 609
   11.23 Converting the Relationship to Aggregation 615
   11.24 Defining the Member Functions Outside Class using Scope Resolution Operator 620
   11.25 Function Overloading and Compile Time Binding 622
   11.26 Local Classes 625
   11.27 Nested Classes 627

12. Constructors and Destructors 652
   12.1 Overview 652
   12.2 Creating Constructors 654
   12.3 Constructor Overloading 659
   12.4 Program to Perform Addition of Point Objects using Constructors: An Example 665
   12.5 Constructor with Default Arguments 668
   12.6 Cloning Objects using Constructor/Copy Constructor 669
   12.7 Allocating Dynamic Memory Inside Constructor 674
   12.8 Destructors in C++ 680
   12.9 Static Members and Static Member Functions 684
## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13.2 Overloading Operators</td>
<td>700</td>
</tr>
<tr>
<td></td>
<td>13.3 Overloading One’s Complement ~ and Minus - Operators: An Example</td>
<td>703</td>
</tr>
<tr>
<td></td>
<td>13.4 Overloading Binary Operators Plus + and Minus -</td>
<td>707</td>
</tr>
<tr>
<td></td>
<td>13.5 Overloading Shorthand Operators: An Example</td>
<td>713</td>
</tr>
<tr>
<td></td>
<td>13.6 Overloading Relational Operators: An Example</td>
<td>716</td>
</tr>
<tr>
<td></td>
<td>13.7 Overloading Increment/Decrement Operators: An Example</td>
<td>720</td>
</tr>
<tr>
<td></td>
<td>13.8 Function Object: Overloading Function Call Operator ()</td>
<td>726</td>
</tr>
<tr>
<td></td>
<td>13.9 Overloading Subscript Operator []</td>
<td>728</td>
</tr>
<tr>
<td></td>
<td>13.10 Overloading Assignment Operator =</td>
<td>731</td>
</tr>
<tr>
<td></td>
<td>13.11 Overloading Type Cast Operator</td>
<td>735</td>
</tr>
<tr>
<td></td>
<td>13.12 Conversion of One User-defined Type to Another</td>
<td>738</td>
</tr>
<tr>
<td></td>
<td>13.13 Creating Global Operator Functions</td>
<td>741</td>
</tr>
<tr>
<td></td>
<td>13.14 Overloading Insertion and Extraction Operator for Student Objects: An Example</td>
<td>745</td>
</tr>
<tr>
<td></td>
<td>13.15 Overloading Operators new and delete</td>
<td>749</td>
</tr>
<tr>
<td></td>
<td>13.16 Overloading operator -&gt;</td>
<td>754</td>
</tr>
<tr>
<td></td>
<td>14.2 Creating a Parent–Child Relationship between Classes</td>
<td>773</td>
</tr>
<tr>
<td></td>
<td>14.3 Access Specifiers in C++: Revisited</td>
<td>776</td>
</tr>
<tr>
<td></td>
<td>14.4 Types of Inheritance</td>
<td>788</td>
</tr>
<tr>
<td></td>
<td>14.5 IS-A and HAS-A Relationship: An Example</td>
<td>793</td>
</tr>
<tr>
<td></td>
<td>14.6 Multi-level Inheritance: Calculator</td>
<td>801</td>
</tr>
<tr>
<td></td>
<td>14.7 Resolving Ambiguity in Multiple Inheritance</td>
<td>808</td>
</tr>
<tr>
<td></td>
<td>14.8 Virtual Base Class</td>
<td>813</td>
</tr>
<tr>
<td></td>
<td>14.9 Function Overriding</td>
<td>821</td>
</tr>
<tr>
<td></td>
<td>14.10 Pointers and Inheritance</td>
<td>824</td>
</tr>
<tr>
<td></td>
<td>14.11 Overriding a Function with Different Return Type</td>
<td>831</td>
</tr>
<tr>
<td></td>
<td>14.12 Virtual Functions and Runtime Polymorphism</td>
<td>833</td>
</tr>
<tr>
<td></td>
<td>14.13 Virtual Tables</td>
<td>843</td>
</tr>
<tr>
<td></td>
<td>14.14 Pure Virtual Functions and Abstract Classes</td>
<td>851</td>
</tr>
<tr>
<td></td>
<td>14.15 static_cast and dynamic_cast</td>
<td>854</td>
</tr>
<tr>
<td></td>
<td>14.16 Constructors and Inheritance</td>
<td>858</td>
</tr>
<tr>
<td></td>
<td>14.17 Working of Constructors with Multiple Inheritance</td>
<td>863</td>
</tr>
<tr>
<td></td>
<td>14.18 Destructors and Inheritance</td>
<td>864</td>
</tr>
<tr>
<td></td>
<td>14.19 Virtual Destructors</td>
<td>867</td>
</tr>
<tr>
<td>15. Input and Output Streams in C++</td>
<td>15.1 Overview</td>
<td>881</td>
</tr>
<tr>
<td></td>
<td>15.2 Types of I/O Streams</td>
<td>882</td>
</tr>
</tbody>
</table>
15.3 Console Input and Output in C++ 883
15.4 Formatted v/s Unformatted I/O Operations 884
15.5 Formatting the Output using Member Functions of Class ios 885
15.6 Formatted I/O using ios Flags 889
15.7 Formatted I/O using Manipulators 893
15.8 Creating your Own Manipulator 894
15.9 Passing Arguments to the Custom Manipulator 895
15.10 Character by Character Unformatted I/O Operations 897
15.11 Line by Line Unformatted I/O Operations 900
15.12 File I/O Operations 902
15.13 Performing Operations on File 904
15.14 Closing the File 914
15.15 File Pointer Manipulation Functions 915

16. Templates in C++ 928
   16.1 Overview 928
   16.2 Function Templates 928
   16.3 Class Templates 933
   16.4 Standard Template Library: One of the Applications of Class Templates 937
   16.5 Implementation of Stack using Linked List: An Example 959
   16.6 Queue using List 964

17. Exception Handling in C++ 969
   17.1 Overview 969
   17.2 Exception Handler in C++ 970
   17.3 throw Keyword 971
   17.4 Examples 971
   17.5 Order of writing Catch Blocks 981
   17.6 Catching and Throwing User-defined Objects 982
   17.7 Program Specifying Throw List 986
   17.8 C++ Built-in Exception Classes 988

ASCII Values 994
List of Keywords in C++ 998
Software Development Life Cycle 999
Bibliography 1002