

Contents

1	Object Oriented Programming	1-9
1.1	Structured Programming	1
1.1.1	Features of Structured Programming	1
1.1.2	Advantages of Structured Programming	2
1.1.3	Disadvantages of Structured Programming	2
1.2	Object Oriented Programming (OOPs)	3
1.2.1	Features of OOPs	4
1.2.2	Why Object Oriented Programming Language is Better?	8
1.2.3	Disadvantages of OOPs	9
1.2.4	Application of OOPs	9
2	Basic C++ Programming	10-36
2.1	History of C++	10
2.2	Introduction to C++	10
2.2.1	Basic Structure of C++ Program	11
2.2.2	Insertion Operator (>>)	12
2.2.3	Extraction Operator (>>)	12
2.2.4	Variable Declaration	13
2.2.4	C++ Program Construction	13
2.3	Type Conversion	15
2.3.2	Typecasting	16
2.4	Array	17
2.4.1	Single/One Dimensional Array	17
2.4.2	Two Dimensional Array	18
2.4.3	Multi Dimensional Array	19
2.5	Pointer	20
2.6	Dynamic Memory Management/New and Delete Operator	21
2.7	Manipulators	22
2.7.1	Non-argument Manipulators	22
2.7.2	Argument Manipulator	23
2.8	Enumeration	24
2.9	cin and cout.....	25
2.10	String	26

2.10.1	Buffer Overflow	27
2.10.2	Reading Embedded Blanks	27
2.10.3	Reading Multiple Lines	28
2.10.3	Copying String	29
2.10.4	Concatenating String	30
	<i>Extra Problems with Solution</i>	31
	<i>Lab Sheet 1</i>	34

3 Functions in C++ 37-52

3.1	Introduction	37
3.2	Why Function	37
3.3	Function Declaration	38
3.4	Function Arguments	39
3.4.1	Passing Argument to Functions	39
3.5	Return Statement and Return Values	41
3.6	Overloading Function	42
3.7	Default Arguments	44
3.8	Inline Functions	45
3.8.1	Reason for the Need of Inline Function	45
3.8.2	What Happens when an Inline Function is Written	46
	<i>Extra Problems with Solution</i>	48
	<i>Lab Sheet 2</i>	51

4 Class and Objects 53-108

4.1	Review of Structure	53
4.1.1	Passing Structure as Function Arguments	55
4.1.2	Limitation of Structure	56
4.2	Introduction to Class and Object	57
4.3	Creating Object	59
4.3.1	Object Persistence and Visibility	60
4.3.2	Memory Allocation for Objects	60
4.4	Member Function	61
4.4.1	Characteristics of Member Functions	61
4.4.2	Defining Member Functions	62
4.4.3	Accessing Class Members	64
4.5	Constructor	66
4.5.1	Characteristics of Constructor	66

4.5.2	Types of Constructor	67
4.6	Destructor	74
4.7	Array of Objects	76
4.8	Objects as Function Arguments	78
4.9	Static Member Function	80
4.10	Friend Function	83
4.11	Friend Class	85
4.11.1	Why Friend Function Breaches the Wall of OOPs? ..	86
4.12	'This' Pointer	86
4.12.1	Using this for Returning Values	87
4.12.2	Using this Pointer to Specify the Memory Address.	88
4.12.3	Accessing Member Data with this	89
	<i>Extra Problems with Solution</i>	90
	<i>Lab Sheet 3</i>	102

5 Operator Overloading 109-150

5.1	Introduction	109
5.2	General Rules for Operator Overloading	111
5.3	Operator Overloading Restrictions	113
5.4	Overloading Unary Operator	113
5.4.1	Returning Values from Operator Function	115
5.5	Binary Operator Overloading	121
5.5.1	Concatenating Strings with Overloaded + Operator	123
5.5.2	Overloading Relational Operator	124
5.5.3	Overloading Equality Operator	126
5.6	Operator Overloading using Friend Function	127
5.7	Type Conversion	129
5.7.1	Conversion from Basic Type to Class Type	130
5.7.2	Conversion from Class Type to Basic Type	131
5.7.3	Conversion from One Class type to Another Class Type	133
	<i>Extra Problems with Solution</i>	136
	<i>Lab Sheet 4</i>	149

6 Inheritance 151-202

6.1	Introduction	151
6.2	Basic Classes and Derived Classes	152
6.3	Access Specifiers	153
6.3.1	Private	153
6.3.2	Protected	155
6.3.3	Public	157
6.4	Constructors in Derived Class	159
6.4.1	Constructor Invocation for Data Member Initialization	164
6.5	Visibility Mode	164
6.5.1	Visibility Mode: Public	165
6.5.2	Visibility Mode: Private	165
6.5.3	Visibility of Class Members	165
6.6	Types of Inheritance	167
6.6.1	Single Inheritance	167
6.6.2	Multilevel Inheritance	168
6.6.3	Hierarchical Inheritance	170
6.6.4	Multiple Inheritance	172
6.6.5	Multipath Inheritance	173
6.6.6	Hybrid Inheritance	174
6.7	Ambiguity in Inheritance	177
6.7.1	Ambiguity in Multiple Inheritances	177
6.7.2	Ambiguity in Multilevel, Multipath and Hybrid Inheritance	178
6.8	Function Overriding	180
6.9	Abstract Base Class	181
6.10	Containership	183
	<i>Extra Problems with Solution</i>	185
	<i>Lab Sheet 5</i>	200

7 Virtual Function and Polymorphism 203-220

7.1	Introduction	203
7.2	What is binding ?	204
7.2.1	Compile Time	204
7.2.2	Run Time	204
7.3	Virtual Function	204
7.3.1	Why we use virtual function?	206

7.4	Virtual Constructors and Destructors	210
7.4.1	Memory	212
7.5	Abstract Classes & Virtual Base Class	212
	<i>Extra Problems with Solution</i>	215
	<i>Lab Sheet 6</i>	218

8 File Handling in C++ 221-246

8.1	Introduction	221
8.2	The Stream Class Hierarchy	222
8.3	Formatted Console I/O Operations	224
8.3.1	Defining Field Width: width()	224
8.3.2	Setting Precision: precision().....	225
8.3.3	Filling and Padding: fill()	226
8.4	Unformatted Console I/O Operations	227
8.5	Opening/Creating File	227
8.5.1	Using Constructor	227
8.5.2	Using Open () Function	229
8.6	Sequential Input Output Operations	230
8.6.1	get () and put () function/character input output operations.....	230
8.6.2	read () and write () Functions/Binary input output operations	232
8.7	Object I/O [Writing and Reading Class Objects]	233
8.8	File Pointers	234
8.9	Working with Multiple File	236
8.10	Error Handling During File Operations	237
8.11	Random Access	238
	<i>Extra Problems with Solution</i>	240
	<i>Lab Sheet 7</i>	245

9 Template in C++ 247-260

9.1	Introduction	247
9.2	Function Templates	248
9.3	Class Templates	250
9.4	Class Templates with Multiple Parameters	255
	<i>Extra Problems with Solution</i>	256
	<i>Lab Sheet 8</i>	260

10 Namespace in C++ 261-268

10.1	Introduction	261
10.2	Using Namespaces	262
10.3	Using Function in Namespace Scope	265
10.4	Using Classes in Namespace	266
10.5	Special Namespace Std	268

11 Analysis and Design Using Object Oriented Techniques 269-284

11.1	Basic Terminologies	269
11.2	CRC Card	270
11.2.1	Components of CRC Card	270
11.2.2	System Requirement	271
11.3	Unified Modeling Language (UML)	272
11.3.1	Goals of UML	272
11.3.2	Why to use UML ?	272
11.4	Use Case	273
11.4.1	Goal of Use Case	273
11.4.2	How to draw use case diagram ?	274
11.4.3	When to use cases diagrams ?	275
11.4.4	Elements of a Use Case Diagram	275
11.5	Class Diagram	278
11.5.1	Goal of Class Diagram	279
11.5.2	How to draw class diagram?	279
11.6	Object Diagram	281
11.6.1	Goal of Object Diagram	281
11.6.2	How to draw object diagram ?	282
11.6.3	Where to use object diagrams?	284
	Bibliography	285