

Contents

<i>Preface</i>	(vii)
----------------------	-------

CHAPTER 1

INTRODUCTION TO COMPUTERS

1.1. INTRODUCTION TO COMPUTERS	1
1.2. HISTORY OF C & C++	3
1.3. DESIGN, DEVELOPMENT AND EXECUTION OF A PROGRAM	3
1.4 TESTING OF PROGRAMS	4
1.4.1 White Box Testing	5
1.4.2 Black Box Testing.....	5
1.4.3 Testing Bench Mark Problems	5
1.5 POINTS TO BE CONSIDERED IN PROGRAMMING	5
1.6 DIFFERENT LEVELS OF PROGRAMMING LANGUAGES	5
1.6.1 Imperative and functional programming languages	6
1.6.1.1 Imperative programming language	6
1.6.2 Functional programming language	7
1.6.3 Interpretation and compilation	8
1.6.3.1 Interpreted language	7
1.6.3.2 Compiled language	7
1.6.3.3 Intermediary languages	8
1.6.4 Some widely used languages	8
1.7. APPROACHES IN PROGRAMMING	8

1.8	TOP-DOWN PROGRAMMING AND STEP-WISE REFINEMENT	10
1.8.1	(a) Advantages of Top down Programming	11
	(b) Disadvantages of Top down programming.....	11
1.9	BOTTOM-UP PROGRAMMING	11
1.9.1	Working of bottom-up programming	11
1.9.2	(a) Advantages of bottom-up programming	12
	(b) Disadvantages of bottom up programming	12
1.10	OPERATING SYSTEMS	13
1.10.1	Types of operating systems	13
1.10.2	Examples of operating systems	15
1.10.3	Components Of an Operating System	18
1.10.4	Modes	19
1.10.5	Networking of Computers	22
1.10.6	Computer Security	23
1.10.7	Operating system user interface.....	24
1.10.8	Diversity of operating systems and portability	25
1.10.9	Black Berry and Black Berry O.S.....	26
1.10.10	C Programming is a subset of C++ programming	27
	COMPREHENSION - 1.....	28
	EXERCISE - 1	30

CHAPTER 2

BASICS OF PROGRAMMING

2.1	INTRODUCTION TO BASICS OF PROGRAMMING	31
2.2	LOGIC FOR PROGRAMMING	31
2.3	ALGORITHM	31
2.3.1	Pseudo code	31
2.3.2	Flow Chart	32
2.4	BASIC CATEGORIES OF OPERATIONS	36
2.5	IMPORTANCE OF 'C' LANGUAGE	37
2.6	IMPORTANCE OF C++ LANGUAGE	38

2.7	SOME OF THE COMPILERS AVAILABLE FOR C AND C++	39
2.7.1	List of compilers for C	40
2.7.2	List of compilers available in C++	40
2.8	STRUCTURE OF A 'C' PROGRAM	40
2.9	PROGRAMMING RULES	43
2.10	EXECUTING THE PROGRAM.....	43
2.11	STEPS TO EXECUTE A C PROGRAM USING TC++ COMPILER.....	45
	COMPREHENSION - 2.....	47
	EXERCISE - 2	48

CHAPTER 3

DECLARATIONS IN C AND C++

3.1.	INTRODUCTION TO C DECLARATIONS	49
3.2.	CHARACTER SET	49
3.2.1	Letters	49
3.2.2	Digits	49
3.2.3	Special Characters	50
3.2.4	White spaces	50
3.2.5	Delimiters	50
3.2.6	Escape sequences or back slash character constants	50
3.3.	C TOKENS / C++ TOKENS	51
3.3.1	Key Words.....	51
3.3.2	Identifiers	52
3.3.3	Constants	52
3.3.3.1	Integer Constants	52
3.3.3.2	Real Constants	54
3.4.	CHARACTER CONSTANTS.....	56
3.4.1	Single Character Constants.....	56
3.4.2	String Constants.....	56
3.5.	SPECIAL SYMBOLS	57
3.6.	OPERATORS	57

3.7. VARIABLES	57
3.7.1 Rules for defining Variables.....	58
3.8. DATA TYPES	59
3.8.1 Primary or Basic or Fundamental Data type.....	59
3.8.1.1 Integer, Float and Char Data types.....	59
3.8.2 User-defined Data Types	61
3.8.2.1 User defined data	61
3.8.2.2 Enumerated datatypes	62
3.8.2.3 Storage Class Data types	63
3.8.2.4 Derived Data Types.....	65
3.8.2.5 Empty Data Set or Void data type in C and C++	66
3.8.2.6 Additional Data Types in C++	66
3.9. DECLARATION AND INITIALIZATION OF VARIABLES	68
3.9.1 Type conversion or type cast	69
3.10. CONSTANT AND VOLATILE VARIABLES	69
3.11. DEFINING SYMBOLIC CONSTANTS.....	70
3.12. MODIFIABILITY	71
3.12.1 Overflow and Underflow of Data	72
3.12.2 Constant Qualifier	72
3.12.3 Reading Data from Keyboard	72
COMPREHENSION - 3.....	74
EXERCISE- 3.....	76

CHAPTER 4

OPERATORS AND EXPRESSIONS

4.1 INTRODUCTION	79
4.2. PRIORITY OR HIERARCHY OR PRECEDENCE OF OPERATORS	79
4.2.1 Use of Different Operators	81
4.2.2 Evaluation of expressions using the priorities of operators.....	87
4.3. TYPE CONVERSIONS IN EXPRESSIONS	89
4.3.1 Operator precedence and associativity	92

4.4	SOME SPECIAL FEATURES OF C++ OVER C :	94
4.5	SOME PROGRAMS IN C	95
4.6	DIFFERENCES BETWEEN OLD AND AND CURRENT VERSIONS OF C++	109
4.6.1	Scope Resolution Operator	109
4.6.2	Differences in I/O statements	110
4.7.	SOME PROGRAMS IN C++ USING TC++	111
	COMPREHENSION - 4	115
	EXERCISE - 4	116

CHAPTER 5

INPUT AND OUTPUT OPERATIONS IN C

5.1.	INTRODUCTION	121
5.1.1	Formatted Input Functions: (without field width specifications)	123
5.1.2	Inputting Integer Variables	123
5.1.3	Inputting Real Variables	124
5.1.4	Inputting Character / String constants	125
5.1.5	Inputting Mixed Variables	125
5.2.	FORMATTED OUTPUT STATEMENTS (WITHOUT FIELD WIDTHS)	126
5.2.1	Outputting Integer Variables	126
5.2.2	Outputting Real variables	127
5.2.3	Outputting Mixed Variables	127
5.3.	FORMATTED INPUT STATEMENTS (WITH FIELD WIDTHS)	128
5.3.1	Inputting Integer numbers	129
5.3.2	Inputting Real Numbers	130
5.3.3	Inputting Characters & Strings	130
5.3.4	Reading Mixed Data Type using scanf ()	131
5.3.5	Detection of Errors in Input ed	131
5.3.6	Commonly used scanf () format codes	132
5.4.	FORMATTED OUTPUT (WITH FIELD WIDTH)	132
5.4.1	Outputting of Integer Numbers	133
5.4.2	Outputting of Real Numbers	134

5.4.3	Outputting of Mixed Variables without field widths	136
5.4.4	Outputting of Strings with field widths	136
5.4.5	Commonly used printf () format codes	137
5.5	ENHANCING THE READABILITY OF OUTPUT	138
5.6	UNFORMATTED INPUT/OUTPUT FUNCTIONS	138
5.6.1	Types of Unformatted Input / Output Functions in C	139
5.7	CHARACTER RECOGNITION USING CODES	143
5.8	COMMONLY USED LIBRARY FUNCTIONS.....	145
5.9	C LIBRARY FUNCTION	145
5.9.1	The Standard C Library Functions	147
5.10	PARTIAL C/C++ FUNCTION LIST	149
5.10.1	Functions Listed Alphabetically	149
5.11.	PROGRAM TO CONVERT DECIMAL NUMBER INTO BINARY	
	NUMBER SYSTEM : DOWNLOAD	150
5.11.1	Logic of This Program	150
5.11.2	Program.....	150
5.11.3	Flow Chart of sub-routine Convert Decimal to Different Number Base	153
	COMPREHENSION - 5	155
	EXERCISE - 5	157

CHAPTER 6

DECISION AND LOOP CONTROL STATEMENTS

6.1.	INTRODUCTION	163
6.1.1	Conditional Operator	163
6.1.2	The Simple if Statement.....	164
6.1.3	Block if statement	164
6.1.4	Nested if statements	165
6.1.5	Simple if else statement	166
6.1.6	Block if else statement	166

6.1.7	Nested ifs or Nested else if or if else if ladder	168
6.1.8	Break statement.....	169
6.1.9	The continue Statement	170
6.1.10	The goto Statement	170
6.1.11	Switch Statement or Switch case statement.....	170
6.2.	THE BREAK STATEMENT.....	172
6.2.1	Exit () Function.....	173
6.3.	LOOPING	174
6.3.1	for LOOP	174
6.4	NESTED for LOOPS.....	178
6.5	WHILE-LOOP	181
6.6	DO-WHILE LOOP.....	183
6.7	The DO-WHILE statement WITH WHILE loop.....	185
6.8	JUMPS IN LOOPS	185
6.8.1	Jumping Out of a Loop using Break Statement	186
6.9	JUMPING WITHIN AND EXITING FROM THE LOOPS WITH GOTO STATEMENT	187
6.10	SKIPPING A PART OF A LOOP USING CONTINUE STATEMENT	189
	COMPREHENSION - 6.....	191
	EXERCISE - 6.....	193

CHAPTER 7

A R R A Y S

7.1	INTRODUCTION	203
7.2	ONE DIMENSIONAL ARRAY.....	203
7.2.1	Single dimensional Array initialisation	204
7.3	TWO-DIMENSIONAL ARRAY	209
7.3.1	Initialization of Two-dimensional arrays.....	210
7.3.2	Two-dimensional sorting.....	215
7.4	MULTIDIMENSIONAL ARRAYS	217
	COMPREHENSION - 7.....	221
	EXERCISE - 7.....	223

CHAPTER 8
FUNCTIONS

8.1	INTRODUCTION	225
8.2	THE RETURN STATEMENT	226
8.2.1	Function with a return statement to return an expression...	229
8.3	CALLING A FUNCTION AND OBTAINING ITS RESULT	230
8.3.1	Call by Value	232
8.4	CALL BY REFERENCE	233
8.5	CATEGORY OF FUNCTIONS	235
8.6	CALLING A FUNCTION WITH NO ARGUMENTS AND NO RETURN VALUES	235
8.7	CALLING FUNCTIONS WITH ARGUMENTS AND NO RETURN VALUES	237
8.8	CALLING FUNCTION WITH ARGUMENTS AND WITH RETURN VALUE	238
8.9	FUNCTION RETURNING MORE VALUES	239
8.10	LOCAL AND GLOBAL VARIABLES	240
8.11	ARGUMENTS OF A FUNCTION	242
8.12	FUNCTION PROTOTYPES	244
8.13	FUNCTIONS AND DECISION STATEMENTS	244
8.14	FUNCTION AND LOOP STATEMENTS	245
8.15	NESTING OF FUNCTIONS	246
8.16	RECURSION	247
8.17	FUNCTIONS WITH ARRAYS AND POINTERS	249
8.18	SPECIAL FUNCTIONS OF C++	252
8.18.1	Default Parameters	253
8.18.2	Inline functions	254
8.18.3	atexit () Function	255
8.19	LIBRARY FUNCTIONS	256
	COMPREHENSION - 8	257
	EXERCISE - 8	262

CHAPTER 9

STORAGE CLASS MODIFIERS

9.1	INTRODUCTION	265
9.2	AUTO VARIABLES OR LOCAL VARIABLES	265
9.3	DYNAMIC STORAGE OR EXTERNAL GLOBAL VARIABLES	266
9.4	STATIC STORAGE OR STATIC VARIABLES	268
9.5	REGISTER VARIABLES	271
	COMPREHENSION - 9	272
	EXERCISE - 9	274

CHAPTER 10

P O I N T E R S

10.1	INTRODUCTION	275
10.2	WHAT IS A POINTER?	275
10.3	POINTER OPERATORS	276
10.4	HOW TO ACCESS A VARIABLE THROUGH ITS POINTER?	277
10.5	POINTER EXPRESSIONS	279
	10.5.1 Pointer Assignments	279
	10.5.2 Arithmetic Operations with Pointers	281
10.6	POINTERS AND ARRAYS	283
10.7	ARRAY OF POINTERS	288
10.8	POINTERS TO POINTERS	289
10.9	POINTERS AND CHARACTER STRINGS	289
	COMPREHENSION - 10	292
	EXERCISE - 10	295

CHAPTER 11

PRE-PROCESSOR DIRECTIVES

11.1	INTRODUCTION	297
11.2	# define DIRECTIVE AND MACRO SUBSTITUTION	297
11.3	UNDEFINING A MACRO	300
11.4	CONDITIONAL COMPILATION USING PRE-PROCESSOR DIRECTIVE	301
11.5	THE #ifndef DIRECTIVE (if not defined)	302
11.6	#if, #else and #elif (else-if)	303
11.7	THE # line DIRECTIVE	304
11.8	THE #error DIRECTIVE	305
11.9	#OPERATOR OR TOKEN PASTING & STRINGIZING	305
11.10	MACRO CONCATENATION WITH THE # # OPERATOR	307
11.11	The # include DIRECTIVE	308
11.12	INLINE DIRECTIVE	308
11.13	THE #pragma SAVEREGS	308
11.14	THE #pragma DIRECTIVE	308
11.15	THE PREDEFINED MACROS IN ANSI AND TURBO-C	310
11.16	STANDARD I/O PREDEFINED STREAMS IN C - stdio.h.....	311
11.17	THE PREDEFINED MACROS IN ctype.h.....	312
	COMPREHENSION - 11	314
	EXERCISE - 11	316

CHAPTER 12

STRUCTURE AND UNION

12.1	INTRODUCTION	319
12.2	FEATURES OF STRUCTURES	319
12.3	DECLARATION AND INITIALIZATION OF STRUCTURES .	320
12.4	STRUCTURE WITHIN STRUCTURE OR NESTED STRUCTURES	324
12.5	ARRAY OF STRUCTURES	328

12.6	STRUCTURES & POINTERS OR POINTER TO STRUCTURE	330
12.7	STRUCTURES AND FUNCTIONS	332
12.8	BIT FIELDS	335
12.9	UNION	337
12.10	ANONYMOUS UNIONS	338
12.11	typedef	339
12.12	ENUMERATED DATA TYPE	340
12.13	CALLING BIOS AND DOS SERVICES	342
12.14	UNION OF STRUCTURES	343
12.15	SCOPE OPERATOR OR SCOPE RESOLUTION OPERATOR :	
	IN C++	344
	COMPREHENSION - 12	346
	EXERCISE - 12	348

CHAPTER 13

FILE OPERATIONS IN C

13.1	INTRODUCTION	351
13.2	STREAMS AND FILE TYPES	351
13.3	STEPS FOR FILE OPERATIONS	351
	13.3.1 Opening of a File	352
13.4	FILE I/O OPERATIONS	362
13.5	BINARY MODE - READ AND WRITE	366
13.6	OTHER FILE FUNCTIONS	367
13.7	SEARCHING ERRORS IN READING / WRITING FILES	368
13.8	USE OF MULTIPLE FILES	370
13.9	LOW LEVEL DISK I/O OPERATIONS	374
13.10	COMMAND LINE ARGUMENTS	376
13.11	APPLICATION OF COMMAND LINE ARGUMENTS	378
13.12	ENVIRONMENT VARIABLES	379
13.13	I/O RE DIRECTION	380
	COMPREHENSION - 13	381
	EXERCISE - 13	383

CHAPTER 14

DYNAMIC MEMORY ALLOCATION - I (LINKED LISTS)

14.1 INTRODUCTION	385
14.1.1 Classification of Data Structures	386
14.2 LINKED LIST	387
14.2.1 Singly Linked List	388
14.2.2 Traversing a singly linked list	392
14.2.3 Addition and Deletion of Elements in a Singly Linked List	394
14.2.4 Insertion in singly linked list	395
14.2.5 Counting of Nodes in a Singly Linked List	407
14.2.6 Deletion of nodes in a Singly linked list	407
14.2.7 Changing the content of an element in a Singly Linked List	412
14.2.8 Reversing of a singly linked list	413
14.3 DOUBLY LINKED LIST (DLL)	415
14.4 MODIFICATION OF DOUBLY LINKED LIST	419
14.4.1 Insertion in a doubly linked list at the beginning	419
14.4.2 Insertion in a doubly linked list at a specified position	419
14.4.3 Deletion of an existing node in a doubly linked list	421
14.5 CIRCULARLY LINKED LIST	423
14.5.1 Singly linked Circular List	423
14.5.2 Doubly linked Circular List	428
14.6. LINKED LIST AS APPLICATIONS OF LINEAR DATA STRUCTURE	429
14.6.1 Polynomial Operation	429
14.6.2 Linked Dictionary	429
COMPREHENSION - 14	430
EXERCISE - 14	431

CHAPTER 15

DYNAMIC MEMORY ALLOCATION - II
(STACKS AND QUEUES)

15.1	INTRODUCTION TO STACKS	433
15.2	IMPLEMENTATION OF STACKS USING LINKED LIST	434
15.3.	STACK RELATED TERMS	435
15.4	OPERATION ON A STACK	436
15.5	IMPLEMENTATION OF A STACK	437
15.6	INTRODUCTION TO QUEUES	445
15.7	REPRESENTATION OF QUEUE	446
15.8	INSERTION AND DELETION OPERATIONS IN A QUEUE	450
15.9	CIRCULAR QUEUES	456
15.9.1	Deletion and insertion in a full circular queue	461
15.10	PRIORITY QUEUE	462
15.10.1	Similarity to queues	462
15.10.2	Implementation of priority queue	463
15.10.3	Effect of different data structures on priority queue	463
15.10.4	Equivalence of priority queues and sorting algorithms	464
15.10.5	Applications of priority queue	465
15.11	DEQUE or DEQUEUE (as Noun), Deque (as verb) and Double-ended Queue	470
15.11.1	Naming conventions of Deque	470
15.11.2	Distinctions and sub-types of Deque	471
15.11.3	Operations of Deque in different languages	471
15.11.4	Implementations of Deque	472
15.11.5	Language support for Deque	472
15.11.6	Complexity of Deque	473
15.11.7	Applications of Deque	473

15.12	REPRESENTATION & OPERATIONS OF ARITHMETIC EXPRESSIONS USING STACK OPERATIONS	473
15.12.1	Infix, prefix, and postfix notations	474
15.12.2	Evaluation of postfix expression	478
15.12.3	Conversion of expression from infix to postfix	480
15.13	RECURSION	493
	COMPREHENSION - 15	494
	EXERCISE - 15.....	496

CHAPTER 16

OPERATIONS ON DATA STRUCTURES (SEARCHING AND SORTING)

16.1	INTRODUCTION	499
16.2	SEARCHING	499
16.2.1	Linear Search or Sequential Search.....	500
16.2.2	Binary Search.....	501
	16.2.2a. Time complexity of binary search	502
	16.2.2b. Comparison of binary search with other techniques	502
16.2.3	Features of Binary Search.....	509
16.2.4	Comparison of Binary Search with other Search Techniques	510
16.3	TREES AND GRAPHS.....	510
16.3.1	Trees.....	510
16.3.2	Binary Trees.....	511
16.3.3	Binary Tree Representation.....	513
16.3.4	Traversing Binary Trees	514
16.4	MINIMUM SPANNING TREE	528
16.4.1	Algorithms to find a minimum spanning tree	531

16.4.2	MST on complete graphs - calculation of approximative expected size	532
16.4.3	Related problems - Types of spanning trees	533
16.5	AVL TREE	534
16.5.1	Basic Operations on an AVL Tree	535
16.5.2	Comparison of AVL to other structures	540
16.6	GRAPHS	540
16.6.1	Prim's algorithm for Traversal of a Graph	546
16.7	TYPES OF TREE SEARCHES	550
16.7.1	Breadth-First Search	550
16.7.2	Depth-First Search	552
16.8	SORTING	557
16.8.1	Exchange Sort (Bubble Sort)	558
16.8.2	Selection Sort	561
16.8.3	Quick Sort	565
16.8.4	Merge sort	571
16.8.5	K-Way Merge Sort	575
16.8.6	What differentiates a K-Way Merge sort from a regular Merge sort?	576
16.8.7	Heap Sort or Tree Sort	579
16.8.8	Insertion Sort (or Simple Insertion Sort)	585
16.8.9	Shell sort.....	589
	COMPREHENSION - 16	593
	EXERCISE - 16.....	596

CHAPTER 17

CLASSES AND OBJECTS IN C++

17.1	INTRODUCTION	590
17.1.1	Difference between Structures and Class	601
17.1.2	Declaration and definition of a simple class	601

17.1.3	Inline Member Functions (Function declared inside the class or outside a class with scope operator):	605
17.1.4	Array of objects	607
17.1.5	Constructor for initializing an object	608
17.1.6	Parameterized Constructors	610
17.1.7	Explicit Constructor	612
17.2	DESTRUCTOR	614
17.3	FRIEND FUNCTIONS	616
17.4	FRIEND CLASSES	618
17.5	DIFFERENT METHODS OF ACCESSION OF OBJECTS	620
17.5.1	Object Assignment	620
17.5.2	Accessing a Private data member from a different object	621
17.6	DYNAMIC MEMORY ALLOCATION IN C++ USING KEY WORDS	622
17.6.1	Const. member function and const. objects	623
17.6.2	Mutable to change const object instance	624
17.6.3	The size of a Class Object	625
17.6.4	Nested Class (Class within another class)	627
17.6.5	The Copy Constructor	629
17.6.6	This Pointer : (this →)	631
17.6.7	Static Class member, Data member & Member Function of a class	632
17.6.8	Classes and structures	635
	COMPREHENSION - 17	637
	EXERCISE - 17	639

CHAPTER 18

INHERITANCE

18.1	INTRODUCTION	641
18.2	SINGLE INHERITANCE	641
18.2.1	Derived Classes	642
18.2.2	Access in Inheritance	646
18.2.3	Constructors and Destructors in inheritance	653

18.2.4	Multilevel inheritance.....	654
18.2.5	Multiple inheritance	658
18.2.6	Virtual base class	661
18.2.7	Passing parameters from derived class constructor to base class constructor	664
	COMPREHENSION - 18	667
	EXERCISE - 18	668

CHAPTER 19

DIFFERENT TYPES OF OVERLOADING IN C++

19.1	INTRODUCTION TO OVERLOADING	669
19.2	TYPES OF OVERLOADING	669
19.2.1	Overloading related to functions - Function Overloading.....	669
19.2.2	Over loading of Function Call Operator ()	673
19.2.3	Constructor overloading.....	674
19.2.4	Operator overloading	676
19.2.4.1	Unary Operator Overloading	678
19.2.4.2	Binary operator overloading	681
19.2.4.3	Overloading Relational Operators	685
19.2.4.4	Overloading Arithmetic Assignment Operators	687
19.2.4.5	Overloading the Assignment Operator	689
19.2.4.6	Overloading Special type of Operator [] ()	690
19.2.4.7	Overloading conversion operators	692
19.2.4.8	Overloading the exponent operator ^	692
19.2.5	Overloading String Concatenation.....	694
19.2.6	Overloading new and delete in memory management	696
19.2.7	Operator Functions as friend for overloaded operator to be commutative :	698
	COMPREHENSION - 19	700
	EXERCISE - 19.....	702

CHAPTER 20

POLYMORPHISM

20.1 INTRODUCTION	703
20.2 COMPILE AND RUNTIME POLYMORPHISM	703
20.3 VIRTUAL FUNCTIONS	706
20.3.1 Pure virtual functions.....	707
20.3.2 Abstract class	709
20.4 TYPES OF DESTRUCTORS	711
20.4.1 Virtual destructor	713
20.4.2 Explicit destructor	714
COMPREHENSION - 20	716
EXERCISE - 20	717

CHAPTER 21

INPUT-OUTPUT OPERATIONS IN C++

21.1 INTRODUCTION	719
21.2 STREAM INSERTION & EXTRACTION	
OPERATORS (<<) & (>>)	720
21.2.1 Stream Insertion Operator for Output (<<).....	720
21.2.2 Stream extraction Operator for Input (>>).....	720
21.2.3 Formatting Output using manipulators	723
21.2.4 Constructing our own manipulators	727
COMPREHENSION - 21	729
EXERCISE - 21	730

CHAPTER 22

FILE OPERATIONS IN C++

22.1 INTRODUCTION	731
22.2 FILE INPUT / OUTPUT IN C++	731
22.2.1 Explicit function for file Closure	733

22.3	INPUT / OUTPUT OPERATIONS ON FILES	734
22.3.1	I/O operations on text files	734
22.3.2	I/O operations on Binary files	739
22.3.3	I/O Operations on Random files	743
22.4	COMMAND LINE ARGUMENTS IN C++	747
22.5	INDEXING DATABASE AND FILES	748
22.5.1	Indexing Data Base	748
22.5.2	Index architecture	749
22.5.3	Operations on Indexing	751
22.5.4	Applications and limitations of Indexing	752
22.6	FILE INDEXING	752
22.6.1	Indexing File Record IDs	752
	COMPREHENSION - 22	754
	EXERCISE - 22	756

CHAPTER 23

SPECIAL FEATURES IN C++

23.1	INTRODUCTION	757
23.2	NAMESPACE	757
23.2.1	Avoiding of namespace using scope operator	757
23.2.2	Use of namespace to avoid use of scope operator	757
23.2.3	Creation of separate namespace in C++ program	758
23.2.4	Use of namespace	759
23.2.5	Switching of namespaces	760
23.2.6	Run-time Type Information (RTTI) using '#include<typeinfo>'	760
23.2.7	Type Casting	761
23.2.8	Four methods of type casting operations in C++	762
23.3	TEMPLATES	766
23.3.1	Types of Templates	767
23.3.2	Faster templates using inline keywords or functions	769
23.3.3	Differences between Class and Function Templates	779

23.4	DEFAULT INITIALIZER IN TEMPLATE TO ONLY CLASS	
	TEMPLATES	781
23.5	EXCEPTION HANDLING	782
23.5.1	Actions of Exception Handling	782
23.5.2	Methods of Exception Handling	783
23.5.3	Handling of different types of Exceptions	784
23.5.4	Constructors and Destructors in Exception Handling	790
23.5.5	Other Exception Specifications	791
	COMPREHENSION - 23	792
	EXERCISE - 23	793

CHAPTER 24

ABSTRACT DATA TYPE (ADT) PROGRAMS

24.1	INTRODUCTION	795
24.2	WHAT IS ADT?	795
24.2.1	Abstract Data Structure	796
24.2.2	Abstract Data Type	796
24.3	SALIENT FEATURES OF ABSTRACT DATA TYPES	798
24.4	WHY ADT?	798
24.5	ADT STACK EXAMPLE	799
24.5.1	Stack ADT	801
24.6	A QUEUE ADT	810
24.7	ADT EMPLOYEE LIST	813
	COMPREHENSION - 24	816
	EXERCISE - 24	817
	REFERENCES	819