Contents

CHAPTER 1 INTRODUCTION TO C 1

1.1 Overview of C 1
1.1.1 Introduction 1
1.1.2 Importance of C language 1
1.1.3 Basic structure of C program 2
1.1.4 Programming style 4
1.1.5 Sample C program 4
1.2 Constants, Variables and Data types 6
1.2.1 Introduction 6
1.2.2 Character set 7
1.2.3 C tokens 8
1.3 Data types 11
1.4 Declaration of variables 13
1.5 Managing I/O operations 17
1.5.1 Introduction 17
1.5.2 Unformatted character reading 17
1.5.3 Formatted I/O 19
1.6 Operators and expressions 23
1.6.1 Introduction 23
1.6.2 Arithmetic operators 23
1.6.3 Increment and Decrement operators 25
1.6.4 Relational operators 26

1.6.5 Logical operators 27
1.6.6 Assignment operators 27
1.6.7 Conditional operators 28
1.6.8 Comma operator 28
1.6.9 Bit-wise operators 29
1.6.10 Cast operator 30
1.6.11 *sizeof* operator 30
1.6.12 Arithmetic expression 31
1.6.13 Operator precedence 32
1.6.14 Type conversion 32
1.7 Decision making, branching, and looping 35
1.7.1 Introduction 35
1.7.2 The `if` statement 35
1.7.3 The `if-else` statement 38
1.7.4 Nested `if-else` statement 39
1.7.5 The `switch` statement 39
1.7.6 The ?: operator 41
1.7.7 The *goto* statement 42
1.7.8 The `while` loop 42
1.7.9 The `do-while` loop 44
1.7.10 The `for` statement 44
1.7.11 The `break` and `continue` statement 46
1.8 Arrays 49
1.8.1 Introduction 49
1.8.2 One-dimensional arrays 49
1.8.3 Two-dimensional arrays 52
1.8.4 Initializing two-dimensional arrays 53
1.9 User defined functions 54
1.9.1 Introduction 54
1.9.2 Advantages of using functions 54
1.9.3 Functions and Multi-functions 55
1.9.4 Return values and their types 56
1.10 Structures 60
1.10.1 Introduction 60
1.10.2 Accessing structure members 61
1.10.3 Initializing a structure 62
1.10.4 Arrays of structures 62
1.10.5 Structures and functions 62
1.10.6 `sizeof` of a structure 63
1.10.7 Union 63

1.10.8 Structure assignment comparison 64

1.11 Pointers 64

1.11.1 Introduction 65

1.11.2 Understanding pointers 65

1.11.3 Declaring and initializing pointer variable 65

1.11.4 Accessing pointer variables 66

1.11.5 Pointer arithmetic 68

1.11.6 Pointers and arrays 70

1.11.7 Character pointers 71

1.11.8 Pointers and functions 71

1.11.9 Pointers and structures 73

1.11.10Pointers and functions 75

1.12 Dynamic Memory Allocation 77

1.12.1 *malloc()* Function 78

1.12.2 *calloc()* Function 79

1.12.3 *realloc()* Function 81

1.12.4 *free()* function 82

1.13 File Management 82

1.13.1 *fopen()* Function 84

1.13.2 *fclose()* Function 85

1.13.3 Stream Functions – "*r*" mode 86

1.13.4 Stream Functions – "*w*" mode 89

1.13.5 Stream Functions – "*a*" mode 90

1.13.6 Binary I/O – *fread()* & *fwrite()* 91

1.13.7 Formatted I/O – *fscanf()* & *fprintf()* 94

1.13.8 Additional Examples 95

1.13.9 Random Access – *fseek()* 96

1.13.10 *ftell()* Function 98

1.13.11 Error handling – *ferror()* & *perror()* 100

1.14 Command Line Arguments 102

1.15 Summary 106

1.16 Exercises 107

CHAPTER 2 INTRODUCTION TO DATA STRUCTURES

2.1 Introduction 109

2.2 Data object 109

2.3 Data type 110

2.4 Primitive data structures 110

2.5 Abstract data type (ADT) 111

2.6 Arrays 112

6.8 Stack as a linked list 211
6.8.1 `Push()` and `Pop()` functions 211
6.9 Queue as a linked list 212
6.9.1 Queue insertion and deletion 213
6.10 Double ended queue (deque) 214
6.11 Ordered linked list 216
6.12 Examples on singly linked list 218
6.13 A comparison between Arrays and linked list 225
6.14 Circular linked list 226
6.14.1 Creating a circular list (front insertion) 227
6.14.2 Inserting a node in a circular list 228
6.14.3 Deleting a node in a circular list 230
6.14.4 Displaying a circular list 232
6.15 Stack as a Circular list 233
6.16 Queue as a Circular list 235
6.17 Circular list with a header node 237
6.18 Doubly linked list 239
6.18.1 Creating a doubly linked list 239
6.18.2 Inserting a node before a key node 241
6.18.3 Deleting a node 242
6.19 Applications 244
6.19.1 Polynomial addition 244
6.19.2 Adding two long positive integers 251
6.20 Linked list with arrays 255
6.21 Summary 257
6.22 Exercises 257

CHAPTER 7 TREES 259

7.1 Introduction 259
7.2 General trees 260
7.3 Binary trees 262
7.4 Properties of binary trees 264
7.5 Representation of binary trees 265
7.6 Tree traversal methods 266
7.6.1 Algorithms and examples 267
7.6.2 From preorder and inorder to binary tree 270
7.7 Binary tree operations and its ADT 271
7.8 Implementation of binary trees 272
7.8.1 Binary Search Tree (BST) 272

7.8.2 Building a BST 273
7.8.3 tree traversals 275
7.8.4 Search 276
7.8.5 Height 277
7.8.6 Size 278
7.8.7 Leaf nodes 278
7.8.8 Finding a min element 279
7.8.9 Copy 280
7.8.10 Delete 281

7.9 Implicit array representation of binary trees 285
7.9.1 `MakeTree()` 287
7.9.2 `Display()` 288

7.10 Priority queues using a heap 289
7.10.1 Heaps 289
7.10.2 Initial heap construction 290
7.10.3 Insertion in MaxHeap 293
7.10.4 Deletion from a MaxHeap 295

7.11 Threaded binary trees 296

7.12 Summary 298

7.13 Exercises 300

CHAPTER 8 SORTING 301

8.1 Introduction 301

8.2 Space and Time complexity 302

8.3 Asymptotic notations 303
8.3.1 Big Oh Notation 304
8.3.2 Omega Notation 305
8.3.3 Theta Notation 306

8.4 Bubble Sort 306

8.5 Quick Sort 309
8.5.1 An Example 310

8.6 Merge Sort 313
8.6.1 The Method 314

8.7 Heap Sort 317

8.8 Selection Sort 320

8.9 Insertion Sort 321

8.10 Shell Sort 322

8.11 Address Calculation Sort 325

8.12 Radix Sort 329

8.13 Summary 334
8.14 Exercises 335

CHAPTER 9 SEARCHING 337

9.1 Introduction 337
9.2 Linear Search 338
9.2.1 Recursive Linear Search 339
9.3 Binary Search 340
9.3.1 Recursive Binary search 342
9.4 Interpolation Search 343
9.5 Hashing 345
9.5.1 Hash functions 346
9.5.2 Collision resolution techniques 347
9.6 Summary 352
9.7 Exercises 352

CHAPTER 10 'C' LAB PROGRAMS 353

10.1 Student records – sequential file 353
10.2 Stack 356
10.3 Infix to Suffix conversion 358
10.4 Postfix expression evaluation 361
10.5 Queue 363
10.6 Circular Queue 365
10.7 Priority Queue 368
10.8 Singly linked list 371
10.9 Stack as linked list 377
10.10 Queue as linked list 379
10.11 Doubly linked list 382
10.12 Binary Search Tree (BST) 387
10.13 (a) Binary Search 389
10.13 (b) Towers of Hanoi 390
10.14 Insertion sort using files 391

Appendix – I Additional Problems 393-406

INDEX 407-410