Contents

List of Figures xiii
List of Tables xv
Preface xvii
Acknowledgments xix

PART A: Fundamentals of Computer

1. Computer-History, Classification and Basic Anatomy 1
   1.1 Generations of Computer 1
   1.2 Classification of Computers 4
   1.3 Basic Anatomy of a Computer 8
   1.4 Von Neumann Architecture 9
   1.5 Memory Classification and Hierarchy 10
   1.6 Input and Output Devices 10

2. Introduction to Number System and Logic Gates 14
   2.1 Introduction 14
   2.2 Base of a Number System 14
   2.3 Reason behind Using Binary Number System 15
   2.4 Conversion among Different Bases 16
   2.5 Similarities between Number Systems of Different Bases 19
   2.6 Addition of Two Numbers 19
   2.7 Signed Binary Numbers 21
   2.8 ASCII (American Standard Code for Information Interchange) 24
   2.9 Logic Gates and Boolean Algebra 25
   2.10 Gates (Logic Gates) 25
   2.11 Basic Gates 26
   2.12 Universal Gates 28

3. Introduction to System Software and Operating Systems 31
   3.1 Introduction to Assembler 31
   3.2 Introduction to Compiler 32
   3.3 Introduction to Operating System 34
Contents

4. Algorithms and Flow Chart 40
  4.1 Flow Chart 40

PART B: Programming in ‘C’

5. Introduction to C 49
  5.1 Generations of Programming Languages 50
  5.2 History of C Language 51
  5.3 Why is C so Popular? 52
  5.4 Position of C in the Generations of Languages 53

6. Constants, Variables and Data Types 58
  6.1 C Character Set 58
  6.2 Constants 58
  6.3 Keywords and Identifiers 61
  6.4 Variables and Data Types 61
  6.5 Storage Type Qualifier 64

7. Operators and Expressions 74
  7.1 Operators in C 74
  7.2 Expressions 82
  7.3 Precedence and Associativity 85

8. Input and Output Statement 98
  8.1 Formatted I/O Functions 98
  8.2 Unformatted I/O Functions 108
  8.3 General Structure of a C Program 109
  8.4 First C Program 110
  8.5 Executing a C Program 111
  8.6 Executing from UNIX Environment 113
  8.7 Developing a Program 113
  8.8 Programming Examples 115

9. Branching Statement 132
  9.1 If Statement 132
  9.2 Conditional Operator 143
  9.3 Switch Statement 144
  9.4 Goto Statement 148
  9.5 Programming Examples 148

10. Loop Statements 171
    10.1 While Statement 172
    10.2 Nested Loop 177
    10.3 Test Your Progress 178
    10.4 For Statement 179
    10.5 Test Your Progress 185
    10.6 Do-while Statement 186
    10.7 Break Statement 187
    10.8 Use of Goto 189
# Contents

1. **Continue Statement** 190
2. **Test Your Progress** 191
3. **Programming Examples** 191

4. **Array** 224
   - **What is an Array?** 224
   - **Declaration of an Array** 224
   - **Initializing Arrays** 225
   - **Accessing Elements of an Array** 228
   - **Sorting** 232
   - **Searching** 240
   - **Two Dimensional Array** 242
   - **Multidimensional Array** 245
   - **Programming Examples** 247

5. **String Handling** 268
   - **String** 268
   - **Overcoming the Limitation with the Help of Scanset** 273
   - **sscanf() and sprintf() Function** 278
   - **Character Handling Functions** 280
   - **String Handling Functions** 289
   - **Two-Dimensional Array of Characters** 294
   - **Programming Examples** 296

6. **Function** 320
   - **Why Function** 320
   - **Calling a Function** 323
   - **Types of Function Arguments** 324
   - **Array as Function Argument** 326
   - **Function Prototype** 328
   - **Passing Multidimensional Array as Function Argument** 331
   - **Storage Class** 332
   - **Multi-file Programs** 337
   - **Recursion** 339
   - **Advantage and Disadvantage of Recursion** 343
   - **Implementation of Some Standard Library Functions** 343
   - **Programming Examples** 345

7. **Pointer** 367
   - **Pointer** 367
   - **Array and Pointer** 371
   - **Pointer Arithmetic** 372
   - **Array vs Pointer** 375
   - **Pointer and Function** 376
   - **Pointer and String** 380
   - **Function Returning Pointer** 381
   - **Pointers and Multidimensional Arrays** 383
   - **Array of Pointers** 384
   - **Dynamic Memory Allocation** 390
<table>
<thead>
<tr>
<th>14.11 Function Pointer</th>
<th>393</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.12 Returning a Two Dimensional Array from Function</td>
<td>394</td>
</tr>
<tr>
<td>14.13 Dangling Pointer</td>
<td>396</td>
</tr>
<tr>
<td>14.14 Implementation of Some Standard Library Functions</td>
<td>397</td>
</tr>
<tr>
<td>14.15 Advantages of Pointer</td>
<td>399</td>
</tr>
<tr>
<td>14.16 Interpreting Complicated Pointer Declarations</td>
<td>400</td>
</tr>
<tr>
<td>14.17 Programming Examples</td>
<td>401</td>
</tr>
<tr>
<td><strong>15. Structure</strong></td>
<td></td>
</tr>
<tr>
<td>15.1 Structure</td>
<td>416</td>
</tr>
<tr>
<td>15.2 Structure Initialization</td>
<td>418</td>
</tr>
<tr>
<td>15.3 Arrays of Structures</td>
<td>422</td>
</tr>
<tr>
<td>15.4 Nested Structures</td>
<td>425</td>
</tr>
<tr>
<td>15.5 Structures and Functions</td>
<td>427</td>
</tr>
<tr>
<td>15.6 Function Returning Structure</td>
<td>428</td>
</tr>
<tr>
<td>15.7 Structures and Pointers</td>
<td>431</td>
</tr>
<tr>
<td>15.8 Bit Fields</td>
<td>434</td>
</tr>
<tr>
<td>15.9 Programming Examples</td>
<td>442</td>
</tr>
<tr>
<td><strong>16. File</strong></td>
<td></td>
</tr>
<tr>
<td>16.1 What is a File?</td>
<td>461</td>
</tr>
<tr>
<td>16.2 Processing a File</td>
<td>462</td>
</tr>
<tr>
<td>16.3 Input-Output Operations on Files</td>
<td>464</td>
</tr>
<tr>
<td>16.4 More File Functions</td>
<td>473</td>
</tr>
<tr>
<td>16.5 Command Line Arguments</td>
<td>477</td>
</tr>
<tr>
<td>16.6 Programming Examples</td>
<td>479</td>
</tr>
<tr>
<td><strong>17. Bitwise Operators</strong></td>
<td></td>
</tr>
<tr>
<td>17.1 Bitwise &amp; Operator</td>
<td>507</td>
</tr>
<tr>
<td>17.2 Bitwise</td>
<td>Operator</td>
</tr>
<tr>
<td>17.3 Bitwise ^ Operator</td>
<td>511</td>
</tr>
<tr>
<td>17.4 ~ Operator</td>
<td>512</td>
</tr>
<tr>
<td>17.5 Left Shift Operator</td>
<td>514</td>
</tr>
<tr>
<td>17.6 Right Shift Operator</td>
<td>515</td>
</tr>
<tr>
<td>17.7 Programming Examples</td>
<td>517</td>
</tr>
<tr>
<td><strong>18. Preprocessors</strong></td>
<td></td>
</tr>
<tr>
<td>18.1 #define Directive</td>
<td>528</td>
</tr>
<tr>
<td>18.2 Macros with Arguments</td>
<td>529</td>
</tr>
<tr>
<td>18.3 Nesting of Macro</td>
<td>531</td>
</tr>
<tr>
<td>18.4 Multiline Macro</td>
<td>531</td>
</tr>
<tr>
<td>18.5 Macro vs. Function</td>
<td>532</td>
</tr>
<tr>
<td>18.6 #undef Directive</td>
<td>532</td>
</tr>
<tr>
<td>18.7 #include Directive</td>
<td>532</td>
</tr>
<tr>
<td>18.8 Conditional Compilation Directive</td>
<td>533</td>
</tr>
<tr>
<td>18.9 Additional Directives</td>
<td>536</td>
</tr>
<tr>
<td>18.10 Predefined Macros</td>
<td>537</td>
</tr>
<tr>
<td>18.11 Programming Examples</td>
<td>538</td>
</tr>
</tbody>
</table>
19. **Linked List**

19.1 Linked List 547
19.2 Advantages of Linked List 547
19.3 Types of Linked List 547
19.4 Implementation of Singly Linked List 548
19.5 Operations on Singly Linked List 550
19.6 Inserting a New Element in a Linked List 553
19.7 Applications of Singly Linked List 568
19.8 Disadvantages of Linked List 572
19.9 Programming Examples 572

**Part C: Technical Questions for Interview**

- **Model Question Set-1**
  Answer to Model Question Set-1 581

- **Model Question Set-2**
  Answer to Model Question Set-2 586

- **Model Question Set-3**
  Answer to Model Question Set-3 591

- **Model Question Set-4**
  Answer to Model Question Set-4 596

- **Model Question Set-5**
  Answer to Model Question Set-5 602

**Appendix:**

- I. Common Questions and Answers 607
- II. ASCII Characters 630
- III. Some Useful Library Functions 635