Basic Computation and Programming with C

Undergraduate students of engineering and computer science will come across detailed coverage of the fundamentals of computation and programming in C language, in this textbook. Essential concepts including operators and expressions, input and output statements, loop statements, arrays, pointers, functions, strings and pre-processor are described in a lucid manner. For better comprehension of the concepts, the book is divided into three parts: Fundamentals of Computer, Programming with 'C' and Technical Questions.

A unique feature of the book is ‘Learn by Quiz – Questions/Answers’, which has questions designed through confidence-based-learning methodology. This helps readers to identify right answers with adequate explanation and reasoning as to why the other options are wrong. Plenty of computer programs and review questions are interspersed throughout the text.

The book can also be used as a self-learning book by beginners in Computer Programming.

Subrata Saha is Head of the Department of Computer Applications, Techno India Hooghly, West Bengal. His areas of interest include object oriented languages, image processing and cryptography.

Subhodip Mukherjee is Head of the Department of Computer Applications, Techno India College of Technology, Kolkata. His areas of interest include object oriented languages, software engineering, computer architecture and database management systems.
Basic Computation and Programming with C

Subrata Saha
Subhodip Mukherjee
To my father Late Kamal Krishna Saha and to my mother Geetasree Saha for what I am today.

— Subrata Saha

To my mother who sacrificed all sorts of entertainment for my education during my school life and to my father Professor S. G. Mukherjee whose addiction to books and simple living inspired me to be a teacher.

— Subhodip Mukherjee
# 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
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

10.9 Continue Statement 190
10.10 Test Your Progress 191
10.11 Programming Examples 191

11. Array 224
11.1 What is an Array? 224
11.2 Declaration of an Array 224
11.3 Initializing Arrays 225
11.4 Accessing Elements of an Array 228
11.5 Sorting 232
11.6 Searching 240
11.7 Two Dimensional Array 242
11.8 Multidimensional Array 245
11.9 Programming Examples 247

12. String Handling 268
12.1 String 268
12.2 Overcoming the Limitation with the Help of Scanset 273
12.3 scanf( ) and sprintf( ) Function 278
12.4 Character Handling Functions 280
12.5 String Handling Functions 289
12.6 Two-Dimensional Array of Characters 294
12.7 Programming Examples 296

13. Function 320
13.1 Why Function 320
13.2 Calling a Function 323
13.3 Types of Function Arguments 324
13.4 Array as Function Argument 326
13.5 Function Prototype 328
13.6 Passing Multidimensional Array as Function Argument 331
13.7 Storage Class 332
13.8 Multi-file Programs 337
13.9 Recursion 339
13.10 Advantage and Disadvantage of Recursion 343
13.11 Implementation of Some Standard Library Functions 343
13.12 Programming Examples 345

14. Pointer 367
14.1 Pointer 367
14.2 Array and Pointer 371
14.3 Pointer Arithmetic 372
14.4 Array vs Pointer 375
14.5 Pointer and Function 376
14.6 Pointer and String 380
14.7 Function Returning Pointer 381
14.8 Pointers and Multidimensional Arrays 383
14.9 Array of Pointers 384
14.10 Dynamic Memory Allocation 390
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.11</td>
<td>Function Pointer</td>
<td>393</td>
</tr>
<tr>
<td>14.12</td>
<td>Returning a Two Dimensional Array from Function</td>
<td>394</td>
</tr>
<tr>
<td>14.13</td>
<td>Dangling Pointer</td>
<td>396</td>
</tr>
<tr>
<td>14.14</td>
<td>Implementation of Some Standard Library Functions</td>
<td>397</td>
</tr>
<tr>
<td>14.15</td>
<td>Advantages of Pointer</td>
<td>399</td>
</tr>
<tr>
<td>14.16</td>
<td>Interpreting Complicated Pointer Declarations</td>
<td>400</td>
</tr>
<tr>
<td>14.17</td>
<td>Programming Examples</td>
<td>401</td>
</tr>
<tr>
<td>15.</td>
<td>Structure</td>
<td>416</td>
</tr>
<tr>
<td>15.1</td>
<td>Structure</td>
<td>416</td>
</tr>
<tr>
<td>15.2</td>
<td>Structure Initialization</td>
<td>418</td>
</tr>
<tr>
<td>15.3</td>
<td>Arrays of Structures</td>
<td>422</td>
</tr>
<tr>
<td>15.4</td>
<td>Nested Structures</td>
<td>425</td>
</tr>
<tr>
<td>15.5</td>
<td>Structures and Functions</td>
<td>427</td>
</tr>
<tr>
<td>15.6</td>
<td>Function Returning Structure</td>
<td>428</td>
</tr>
<tr>
<td>15.7</td>
<td>Structures and Pointers</td>
<td>431</td>
</tr>
<tr>
<td>15.8</td>
<td>Bit Fields</td>
<td>434</td>
</tr>
<tr>
<td>15.9</td>
<td>Programming Examples</td>
<td>442</td>
</tr>
<tr>
<td>16.</td>
<td>File</td>
<td>461</td>
</tr>
<tr>
<td>16.1</td>
<td>What is a File?</td>
<td>461</td>
</tr>
<tr>
<td>16.2</td>
<td>Processing a File</td>
<td>462</td>
</tr>
<tr>
<td>16.3</td>
<td>Input-Output Operations on Files</td>
<td>464</td>
</tr>
<tr>
<td>16.4</td>
<td>More File Functions</td>
<td>473</td>
</tr>
<tr>
<td>16.5</td>
<td>Command Line Arguments</td>
<td>477</td>
</tr>
<tr>
<td>16.6</td>
<td>Programming Examples</td>
<td>479</td>
</tr>
<tr>
<td>17.</td>
<td>Bitwise Operators</td>
<td>507</td>
</tr>
<tr>
<td>17.1</td>
<td>Bitwise &amp; Operator</td>
<td>508</td>
</tr>
<tr>
<td>17.2</td>
<td>Bitwise</td>
<td>Operator</td>
</tr>
<tr>
<td>17.3</td>
<td>Bitwise ^ Operator</td>
<td>512</td>
</tr>
<tr>
<td>17.4</td>
<td>~ Operator</td>
<td>514</td>
</tr>
<tr>
<td>17.5</td>
<td>Left Shift Operator</td>
<td>515</td>
</tr>
<tr>
<td>17.6</td>
<td>Right Shift Operator</td>
<td>517</td>
</tr>
<tr>
<td>17.7</td>
<td>Programming Examples</td>
<td>518</td>
</tr>
<tr>
<td>18.</td>
<td>Preprocessors</td>
<td>528</td>
</tr>
<tr>
<td>18.1</td>
<td>#define Directive</td>
<td>528</td>
</tr>
<tr>
<td>18.2</td>
<td>Macros with Arguments</td>
<td>529</td>
</tr>
<tr>
<td>18.3</td>
<td>Nesting of Macro</td>
<td>531</td>
</tr>
<tr>
<td>18.4</td>
<td>Multiline Macro</td>
<td>531</td>
</tr>
<tr>
<td>18.5</td>
<td>Macro vs. Function</td>
<td>532</td>
</tr>
<tr>
<td>18.6</td>
<td>#undef Directive</td>
<td>532</td>
</tr>
<tr>
<td>18.7</td>
<td>#include Directive</td>
<td>532</td>
</tr>
<tr>
<td>18.8</td>
<td>Conditional Compilation Directive</td>
<td>533</td>
</tr>
<tr>
<td>18.9</td>
<td>Additional Directives</td>
<td>536</td>
</tr>
<tr>
<td>18.10</td>
<td>Predefined Macros</td>
<td>537</td>
</tr>
<tr>
<td>18.11</td>
<td>Programming Examples</td>
<td>538</td>
</tr>
</tbody>
</table>
19. **Linked List** 546
  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** 581
  Answer to Model Question Set-1 584

- **Model Question Set-2** 586
  Answer to Model Question Set-2 589

- **Model Question Set-3** 591
  Answer to Model Question Set-3 594

- **Model Question Set-4** 596
  Answer to Model Question Set-4 599

- **Model Question Set-5** 602
  Answer to Model Question Set-5 605

**Appendix:**

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

Fig. 1.1 ENIAC 2
Fig. 1.2 Transistor 3
Fig. 1.3 Components of a computer 8
Fig. 1.4 Common bus architecture 9
Fig. 1.5 Keyboard 11
Fig. 1.6 USB and PS/2 ports 11
Fig. 1.7 Scanner 12
Fig. 1.8 Microphone 12
Fig. 1.9 Printers 13
Fig. 2.1 Addition process 20
Fig. 2.2 Sequential circuit 25
Fig. 2.3 NAND as universal gate 28
Fig. 3.1 Process for producing executable file 32
Fig. 3.2 Parse tree of a sentence 33
Fig. 3.3 A sample window 36
Fig. 4.1 Flowchart of finding maximum of three numbers 41
Fig. 8.1 Steps of compilation and execution 112
Fig. 9.1 Flowchart of if statement 133
Fig. 9.2 Flowchart of if-else statement 134
Fig. 9.3 Flow chart to find the largest number among three inputted numbers 137
Fig. 10.1 Entry controlled loop 171
Fig. 10.2 Exit controlled loop 172
Fig. 10.3 Flow chart to illustrate while statement 173
Fig. 11.1 Allocation of memory for character array 226
Fig. 11.2 Allocation of memory for integer array 227
Fig. 11.3 Conceptual view of a two dimensional array 242
Fig. 11.4 Accessing elements in a two dimensional array 243
Fig. 14.1 Memory map of variable allocation 368
Fig. 14.2 Address calculation for the elements of 2D array 383
Fig. 14.3 Memory map of above example 385
Fig. 14.4 Memory map of array of pointer to string 386
Fig. 15.1 Memory map of structure variable 417
Figures

Fig. 15.2 Array of structure 423
Fig. 15.3 Initialized array of structure s 423
Fig. 15.4 Sharing of memory location by the members of a union 437
Fig. 17.1 << operation 516
Fig. 17.2 >> operation 517
Fig. 19.1 Different type of linked list 548
Fig. 19.2 Representation of a singly linked list in memory 550
Fig. 19.3 Inserting a node at the beginning of a list 554
Fig. 19.4 Inserting a node at the end of a list 556
Fig. 19.5 Inserting a node after 2nd node 557
Fig. 19.6 Inserting a node before 3rd node 557
Fig. 19.7 Deletion of first node from a list 559
Fig. 19.8 Deletion of last node from a list 560
Fig. 19.9 Deletion of an intermediate node from a list 562
Fig. 19.10 Representation of a polynomial using linked list 568
# Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Powers of 2</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>Binary values of decimal numbers</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Similarity of different base</td>
<td>19</td>
</tr>
<tr>
<td>2.4</td>
<td>ASCII values</td>
<td>24</td>
</tr>
<tr>
<td>3.1</td>
<td>Some DOS commands</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>Some UNIX commands</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>Flowchart symbols</td>
<td>42</td>
</tr>
<tr>
<td>6.1</td>
<td>C character set</td>
<td>58</td>
</tr>
<tr>
<td>6.2</td>
<td>Keywords in C</td>
<td>61</td>
</tr>
<tr>
<td>6.3</td>
<td>Basic data types in C</td>
<td>62</td>
</tr>
<tr>
<td>6.4</td>
<td>Size and range of data types on 16 bit machine</td>
<td>62</td>
</tr>
<tr>
<td>7.1</td>
<td>Arithmetic operators and their role</td>
<td>75</td>
</tr>
<tr>
<td>7.2</td>
<td>Shorthand assignment operators and their equivalent statement</td>
<td>77</td>
</tr>
<tr>
<td>7.3</td>
<td>Relational operators</td>
<td>79</td>
</tr>
<tr>
<td>7.4</td>
<td>Logical operators</td>
<td>80</td>
</tr>
<tr>
<td>7.5</td>
<td>Bitwise operators</td>
<td>81</td>
</tr>
<tr>
<td>7.6</td>
<td>Conversion rules</td>
<td>83</td>
</tr>
<tr>
<td>7.7</td>
<td>The precedence of operators and their associativity</td>
<td>86</td>
</tr>
<tr>
<td>8.1</td>
<td>Escape sequences</td>
<td>99</td>
</tr>
<tr>
<td>8.2</td>
<td>Conversion specifiers</td>
<td>100</td>
</tr>
<tr>
<td>8.3</td>
<td>Minimum width specifiers</td>
<td>102</td>
</tr>
<tr>
<td>8.4</td>
<td>Flag specifiers</td>
<td>104</td>
</tr>
<tr>
<td>8.5</td>
<td>Conversion specifier that are used with flag specifiers</td>
<td>104</td>
</tr>
<tr>
<td>8.6</td>
<td>Size modifiers with their conversion specifier</td>
<td>105</td>
</tr>
<tr>
<td>8.7</td>
<td>Examples of size modifiers</td>
<td>106</td>
</tr>
<tr>
<td>16.1</td>
<td>File opening mode</td>
<td>462</td>
</tr>
<tr>
<td>16.2</td>
<td>Reference positions</td>
<td>474</td>
</tr>
<tr>
<td>17.1</td>
<td>Bitwise operators</td>
<td>507</td>
</tr>
<tr>
<td>17.2</td>
<td>Bitwise &amp; operation on individual bits</td>
<td>508</td>
</tr>
<tr>
<td>17.3</td>
<td>Illustration of bitwise &amp; operation showing bit pattern</td>
<td>508</td>
</tr>
<tr>
<td>17.4</td>
<td>Bitwise</td>
<td>operation on individual bits</td>
</tr>
<tr>
<td>17.5</td>
<td>Illustration of bitwise</td>
<td>operation showing bit pattern</td>
</tr>
</tbody>
</table>
Tables

Table 17.6  Bitwise ^ operation on individual bits  512
Table 17.7  Illustration of bitwise ^ operation showing bit pattern  513
Table 17.8  Bitwise ~ operation on individual bits  515
Table 17.9  Illustration of left shift operation showing bit pattern  516
Table 17.10 Illustration of right shift operation showing bit pattern  517
Table 18.1  Predefined macros  537
Preface

The C programming language is one of the most, academically as well as industrially, important programming languages in the world. It was unveiled in 1972 and since then, with gradual enhancements and enrichments, it has successfully established itself as a powerful language for programming microcontrollers, operating systems and various commercially significant software packages. Having unique features like block structure, stand alone functions and a rich set of keywords with very few restrictions, it is aptly regarded as the ‘Mother Language’ among programming languages. Hence it is necessary for every programmer to learn and, often, use C.

About the Book

Today, in any corner of the world, there is a necessity to learn computer programming for people to survive in any industry. All students do not possess technical background and therefore find learning C programming difficult. Keeping a strong focus on industrial requirements and the limitations of students from non-technical backgrounds, we have written this new book on C programming to enable students of non-technical as well as technical backgrounds learn this marvelous programming language in a completely new way.

This book is completely different from other popular C programming books available in the market. It teaches programming to someone who wants to learn how to program in C for the first time in his/her life with no programming knowledge at all. In other words, this book on C programming is for absolute beginners of programming but at the same time it outclasses several other C programming books in the market with its coverage. Although targeted at complete beginners, this book covers many advanced concepts in C programming and enables a non-programmer to develop into a competitive C programmer ready to face job interviews on C.

Structure of the Book

- This book is written presuming that the reader is not strong in English. All the explanations in each chapter are offered in simplest English to enable any reader to correctly understand a concept.
Each chapter contains appropriate examples. It ends with a ‘Learn by Quiz – Questions/Answers’, having multiple choice questions with their respective answers (with clarifications) to ensure understandability.

In order to make the presentation visually interactive for students, neat labeled diagrams are provided wherever necessary. For each topic, the explanations are clear and concise, avoiding verbosity as much as possible.

Each programming topic is presented with an exercise set consisting of a large number of solved C programs. Each such solved program is presented with WHYs and HOWs for each new or next statement of a program.

This book also includes a rich collection of commonly asked as well as most probable, new C programming interview solved questions and programs with step-by-step explanations to enable a student to be successful in an interview.

Students, we have prepared this book for you using a unique teaching approach based on our academic learnings so that it looks least intimidating and most interesting. We are sure that after studying this new book on C programming, those of you for whom C programming has been a source of sorrow will now find in it joy and fun. Though every attempt has been made to avoid errors, we will be grateful to our readers if they bring to our notice any oversight they find. If the content design and organization of this book meets all the expectations and requirements of our students, then only can this book be considered a worthwhile accomplishment.
Acknowledgments

I thank my, beloved, students for their constant motivation which made me write this book despite the availability of many best-selling C programming books in the market. My students are extremely fond of and fascinated by my ‘teaching from the ground up’ style and their demand inspired me to author this amazingly simple C book which is aimed at non-programmers or absolute beginners. My heartiest thanks to them.

I would like to thank T. K. Ghosh, Chief Executive Director of Techno India Group for providing me the platform to teach so many young minds.

I want to thank M. K. Chakraborty for his constant support and inspiration.

I would also like to thank Subir Hazra, my friend and ex-colleague, who helped us in preparing the ‘Learn by Quiz - Questions/Answers’ section.

Most important, heartiest thanks to my family specially my wife Sriparna Saha and my beloved daughter Shreya for their support, encouragement, quite patient and unconditional love which helped, conclude the book.

Finally, I would like to thank all the reviewers of this book for their critical comments and suggestions. I convey my gratitude to Rachna Sehgal and the entire editing team at Cambridge University Press, India for their great work.

Subrata Saha

Thanks to T. K. Ghosh (Chief Executive Director) who found in me suitability to be the head of three departments of a Techno India Group college. He gave me the opportunity to prove myself, to be a good teacher to thousand of students.

Thanks to my teacher Professor Ranjan Dasgupta (Head, Computer Science, NITTTR-Kolkata) who gave me the confidence to write a book.

Thanks to my ex-colleague Professor Sumana Chakraborty who participated actively in discussions to freeze the table of contents of this book.

Subhodip Mukherjee