



# PROGRAMMING IN C

[ IN ENGLISH LANGUAGE ]

By  
Jignesh Shah

**Edition** : 1<sup>st</sup> Edition : 2006  
**ISBN** : 8185594635, 9788185594637  
**Size** : 170 mm x 240 mm  
**Binding** : Paperback  
**Pages** : 448 + 16 = 464



₹ 150.00 **BUY**

## ABOUT THE BOOK

C language is very important in Computer Programming. This book is divided into 12 chapters and 5 Appendix. So that students can get detailed, systematic and step by step knowledge of the subject. In each chapter all the points are explained in simple language with illustrations.

*Salient features of the book are as follows:*

- \* 527 Examples (including programs with output)
- \* 79 Important tables
- \* 376 Exercise Questions.

watch, output, trace into, step over, getch( ), clrscr( ), gotoxy( ), kbhit( ), delay( ), time structure, merits and demerits of string input library function are discussed for the first time in this book.

In addition to this, 'C' fundamentals, operators, input-output library functions, loops, string handling library functions, pointers, multifile programming, file handling functions are discussed in detail in easy to understand style.

In Chapter 10 Object oriented programming is discussed in detail and in Chapter 11 special programs are given. By studying these programs, programmer can become expert in 'C' language.

*In this book following Appendices are explain in detail:*

- \* Appendix A – Algorithm and Flow-chart,
- \* Appendix B – Bit level Programme
- \* Appendix C – ASCII Character Set
- \* Appendix D – Operators precedence table and
- \* Appendix E – Library Functions.

This book is prepared for both learners as well as for experienced programmers. This book is very useful for Degree and Diploma students of Engineering, M.C.A., P.G.D.C.A., B.C.A., B.Sc., I.T.I., Commerce, Management and students learning Programming Languages in private Institutes.

## CONTENT

- 1 : 'C' FUNDAMENTALS
  - 2 : OPERATORS AND INPUT-OUTPUT FUNCTIONS
  - 3 : STATEMENT, BRANCHING AND LOOPING
  - 4 : 'C' FUNCTIONS – USER DEFINED FUNCTION
  - 5 : STORAGE CLASSES
  - 6 : ARRAY
  - 7 : POINTERS
  - 8 : STRUCTURES AND UNIONS
  - 9 : WORKING WITH FILES
  - 10 : LINK LIST
  - 11 : INTRODUCTION TO OBJECT ORIENTED PROGRAMMING
  - 12 : SPECIAL PROGRAMS
- APPENDIX A : FLOWCHART AND ALGORITHM  
APPENDIX B : BIT OPERATORS  
APPENDIX C : ASCII CHARACTER SET TABLE  
APPENDIX D : OPERATOR PRECEDENCE TABLE  
APPENDIX E : STANDARD LIBRARY FUNCTIONS  
INDEX

Checklist

**PROGRAMMING IN C**  
**DETAILED CONTENTS**

**Chapter 1 'C' FUNDAMENTALS**

- 1-1. Introduction
- 1-2. Features of 'C'
  - (1) General purpose language
  - (2) Structured programming language
  - (3) Middle level language
  - (4) Portable language
  - (5) Free style language
  - (6) Case sensitive language
- 1-3. Advantages of 'C'
- 1-4. Structure of a 'C' program
  - (1) #include
  - (2) Comments
  - (3) main( )
  - (4) Braces
  - (5) printf
  - (6) Semicolon
- 1-5. getch( ) and clrscr( )
- 1-6. Watch, output and trace into
  - (1) Setting watches
  - (2) Trace into
  - (3) Output screen
- 1-7. Writing, compiling and running a 'C' program
- 1-8. Character set
  - (1) Letters
  - (2) Digits
  - (3) Special characters
  - (4) White space characters
- 1-9. Keywords
- 1-10. Identifiers
- 1-11. Data-types
  - (1) void
  - (2) Integral data-type
  - (3) Floating-point types
- 1-12. Constants
  - (1) Integer constants
  - (2) Floating-point constant
  - (3) Character constant
  - (4) String constant
- 1-13. Symbolic constants
- 1-14. Escape sequence
- 1-15. Variables and their declarations
  - Declaration of variable
- 1-16. Short-cut keys
  - (1) Cursor movement commands
  - (2) Block commands
  - (3) Other important commands

**Chapter 2 OPERATORS AND INPUT-OUTPUT FUNCTIONS**

- 2-1. Introduction
  - (1) Operators on the basis of number of operands
  - (2) Functionality of operators
- 2-2. Arithmetic operators
  - Precedence of Arithmetic operators
- 2-3. Unary operators
  - (1) Unary Minus
  - (2) Increment operator and decrement operator
  - (3) Size of operator
  - Precedence of Unary operators
- 2-4. Relational operators
  - Precedence of Relational operator
- 2-5. Equality operators
  - Precedence of equality operators
- 2-6. Logical operators
  - (1) AND Operator
  - (2) OR Operator
  - (3) NOT Operator
  - Precedence of logical operators

- 2-7. Assignment operators
  - (1) = : assign the value
    - Difference between Assignment Operator (=) and Equality Operator (==)
  - (2) +=, -=, \*=, /= and %= : add then assign, subtract then assign, multiply then assign, divide then assign and evaluate remainder then assign
- 2-8. Conditional operator
- 2-9. Precedence of operators
- 2-10. Library functions
  - (1) Input/Output Operation
  - (2) For Operation on character
  - (3) For operation on string
  - (4) For Arithmetic operations
- 2-11. Type conversion
  - (1) Implicit conversion
  - (2) Explicit conversion
- 2-12. Input-output functions
- 2-13. getchar function
- 2-14. putchar function
- 2-15. scanf function
- 2-16. printf function
- 2-17. getstring function
- 2-18. putstring function
- 2-19. field - width
  - Rules for field-width
- 2-20. Typical programs
- Exercises

**Chapter 3 STATEMENT, BRANCHING AND LOOPING**

- 3-1. Introduction
  - (1) Condition checking
  - (2) Body of the loop
- 3-2. if statement
  - (1) Simple if statement
  - (2) if.....else statement
  - (3) Compound if statement
  - (4) Nested if.....else statement
  - (5) else if ladder
- 3-3. For loop
  - (1) Initialization
  - (2) Condition checking
  - (3) Increment / Decrement
  - Nesting of for loops
  - Programs of nested for loop
- 3-4. While loop
  - Nesting of while loop
- 3-5. Do-while loop
  - (1) for and do-while
  - (2) while and do-while
  - Nesting of do - while loop
- 3-6. The switch statement
- 3-7. Break statement
- 3-8. exit function
- 3-9. Continue statement
- 3-10. Goto statement
- 3-11. gotoxy function
- 3-12. Special programs
- Exercises 3

**Chapter 4 'C' FUNCTIONS – USER DEFINED FUNCTION**

- 4-1. Introduction
- 4-2. Needs/importance/advantages
- 4-3. Function Definition
  - (1) First line
  - (2) Body of the function
- 4-4. Return statement
- 4-5. Calling a function

**PROGRAMMING IN C**  
**DETAILED CONTENTS**

- 4-6. Function prototyping
  - (1) No argument, No returntype
  - (2) No argument, With returntype
  - (3) With argument, Without returntype
  - (4) With argument, With returntype
- 4-7. Passing arguments to a function
  - (1) Call by value
  - (2) Call by reference
- 4-8. Nesting of functions
- 4-9. Recursion
- 4-10. Typical Programs
- 4-11. Macros
- Exercises 4

**Chapter 5 STORAGE CLASSES**

- 5-1. Introduction
- 5-2. Types of storage classes
- 5-3. Automatic variables
- 5-4. External variable
  - Difference between external variable definition and declaration
  - Inherent dangers of external variables
- 5-5. Static variable
- 5-6. Register variable
- 5-7. Multi-file Programming
  - Rules associated with the use of functions in multi-file programming
  - Variables in multi-file programs
  - How to run multi-file programs
  - Exercises 5

**Chapter 6 ARRAY**

- 6-1. Introduction
- 6-2. What is an array?
- 6-3. Defining one-dimensional array
- 6-4. Initializing arrays
- 6-5. Processing an array
- 6-6. Passing array to function
- 6-7. Two-dimensional arrays
- 6-8. Initializing two-dimensional arrays
- 6-9. Array and strings
- 6-10. Declaration and initialization of strings
- 6-11. How to input strings?
- 6-12. How to print strings to the screen?
  - (1) %s
  - (2) putchar( )
  - (3) puts( )
- 6-13. Finding length of a string
- 6-14. Copying strings
- 6-15. Comparison of two strings
- 6-16. Reversing a string
- 6-17. Concatenating strings
- 6-18. String handling functions
  - (1) String length functions
  - (2) String copy functions
  - (3) String compare functions
  - (4) String reverse functions
  - (5) String concatenate function
  - (6) String lower functions
  - (7) String upper functions
  - (8) String set function
  - (9) Bytes swapping function
- 6-19. Typical programs
- Exercises 6

**Chapter 7 POINTERS**

- 7-1. Introduction
- 7-2. What is pointer?
- 7-3. Declaration and initialization of pointers
  - (1) char \*ch
  - (2) int \*number
  - (3) float \*fptr
- 7-4. '&' and '\*' operators
  - (1) &
  - (2) \*
- 7-5. Pointer and one-dimensional arrays
- 7-6. Pointer and two-dimensional array
- 7-7. Dynamic memory allocation
  - Allocating a block of memory
  - Freeing the used space
  - Modifying block size
- 7-8. Operations on pointers
- 7-9. Pointer and functions
- 7-10. Functions returning pointers
- Exercises 7

**Chapter 8 STRUCTURES AND UNIONS**

- 8-1. Introduction
- 8-2. Structure declaration
- 8-3. Dot operator
- 8-4. How to give values to structure members?
- 8-5. Structure initialization
- 8-6. Using unary operators with structures
- 8-7. Using assignment operator with structure
  - (1) Assigning complete structure variable to another structure variable
  - (2) Assigning values in the individual members of the structure variable
- 8-8. Array of structures
- 8-9. Arrays within structures
- 8-10. Structures within structures – nested structure
- 8-11. Structures and functions
- 8-12. Structures and pointers
  - How pointers to structure variables declare?
  - How a structure is accessed using its pointer?
- 8-13. typedef feature of 'C'
- 8-14. Unions
- 8-15. Typical programs
  - (1) Structure time
  - (2) gettime( )
  - (3) kbhit( )
  - (4) delay( )
- Exercises 8

**Chapter 9 WORKING WITH FILES**

- 9-1. Introduction
- 9-2. Opening a file
- 9-3. Closing a data-file
- 9-4. File input/output functions
  - (1) putc function
  - (2) getc function
  - (3) fputc function
  - (4) fgetc function
  - (5) fputs function
  - (6) fgets function
  - (7) putw function
  - (8) getw function
  - (9) fwrite function
  - (10) fread function
  - (11) fprintf function
  - (12) fscanf function

## PROGRAMMING IN C DETAILED CONTENTS

- 9-5. Opening a file in append mode
  - 9-6. A program to copy a file
  - 9-7. Error handling during file I/O operation
    - (1) feof( ) function
    - (2) ferrord( ) function
  - 9-8. Movement of file pointer
    - (1) fseek function
    - (2) ftell( ) function
    - (3) rewind function
- Exercises 9

### **Chapter 10 LINK LIST**

- 10-1. Introduction
- 10-2. Linked list
- 10-3. Types of linked list
  - Two way list
  - Circular linked list
- 10-4. Basic list operation
- 10-5. Creating a linked list
- 10-6. Traversing a linked list
- 10-7. Searching a linked list
- 10-8. Insertion into linked list
  - 10-8-1. Insertion at first node in the list
  - 10-8-2. Insertion at last in the list
  - 10-8-3. Insertion at middle in the list
    - Algorithm for inserting node at middle of the list
    - Deletion of node at first in link list
    - Doubly linked linear lists
    - Steps for Algorithm

### **Chapter 11 INTRODUCTION TO OBJECT ORIENTED PROGRAMMING**

- 11-1. Introduction
- 11-2. Needs for object oriented programming
  - (1) Procedural languages
  - (2) Division into functions
  - (3) Problems with structured programming
  - (4) Data undervalued
  - (5) Relationship to the real world
  - (6) New data types
- 11-3. Object Oriented Approach
- 11-4. Characteristics of OOP language
  - (1) Classes & objects
  - (2) Data abstraction and encapsulation
  - (3) Inheritance
  - (4) Polymorphism and overloading

- (5) Creating new data types
- 11-5. Something about C++
  - (1) cout – output operator
  - (2) cin – input operator
- 11-6. Examples on C++  
Exercises 11

### **Chapter 12 SPECIAL PROGRAMS**

#### **Appendix A FLOWCHART AND ALGORITHM**

- A-1. Introduction
- A-2. Algorithm
- A-3. Characteristics
- A-4. Flowchart
- A-5. Flowchart symbols
  - (1) Input & output
  - (2) Processor
  - (3) Terminal
  - (4) Decision maker
  - (5) Connector
- A-6. Computer flowcharting symbols
- A-7. Arithmetic operators and comparators
- A-8. Typical flowcharts

#### **Appendix B BIT OPERATORS**

- B-1. Introduction
- B-2. Bitwise operators
- B-3. Bitwise logical operators
  - (1) Bitwise AND operator
  - (2) Bitwise OR operator
  - (3) Bitwise exclusive OR
- B-4. Bitwise shift operators
  - (1) Left shift operator
  - (2) Right shift operators
- B-5. Bitwise complement operator

#### **Appendix C ASCII CHARACTER SET TABLE**

- C-1. Control character set
- C-2. Printing character set
- C-3. Extended ASCII character set

#### **Appendix D OPERATORS PRECEDENCE TABLE**

#### **Appendix E STANDARD LIBRARY FUNCTIONS**

- |              |                 |
|--------------|-----------------|
| E-1. alloc.h | E-7. graphics.h |
| E-2. conio.h | E-8. math.h     |
| E-3. ctype.h | E-9. stdio.h    |
| E-4. dir.h   | E-10. stdlib.h  |
| E-5. dos.h   | E-11. String.h  |
| E-6. io.h    | E-12. time.h    |

#### **Index**