

GUIDE TO LEARN C PROGRAMMING

Prof. Melwin D Souza



GUIDE TO LEARN C PROGRAMMING

FIRST EDITION

AUTHOR

Prof. Melwin D Souza

B.E, M.Tech.,LMISTE, (PhD)

Associate Professor, Department of Computer Science and Engineering

MOODLAKATTE INSTITUTE OF TECHNOLOGY KUNDAPURA



(SCIENTIFIC INTERNATIONAL PUBLISHING HOUSE)

Title of the Book: Guide to Learn C Programming

Edition: First - 2022

Copyrights © Author

Disclaimer

The contents given in this textbook are mainly meant for learning C programming from the beginning. The author is solely responsible for the contents published in this textbook he publishers or editors do not take any responsibility for the same in any manner. Errors, if any, are purely unintentional and readers are requested to communicate such errors to the editors or publishers to avoid discrepancies in the future.

ISBN: 978-93-5625-580-7

MRP: 630/-

PUBLISHER & PRINTER: Scientific International Publishing House

Contact: +917019991025

Website: www.sipinternationalpublishers.com

Preface

C is one of the most powerful programming languages in computer science and it is currently the premier language for software developers. It is very important to understand the flow of control in programming and learn any high-level programming language.

This book highlights the skills you will need to do real-world programming. This book is mainly focused to learn programming from the beginning. All the concepts are discussed with the help of examples and programs. Based on the requirement, more programming examples are solved on standard problems.

Simple language and scenarios are used to explain the topics for better interpretation. Special care is taken to place and to maintain the continuity of topics. In this textbook, 32-bit system architecture is considered for an explanation.

Acknowledgment

First and foremost, I would like to praise and thank God, the Almighty, who has granted me countless blessings, knowledge, and opportunity to write this book.

Apart from my efforts, the success of this textbook depends largely on the encouragement and guidelines of many others.

My sincere thanks to the management of Moodlakatte Institute of Technology Kundapura for providing continuous support and motivation during the work.

I would like to express my special thanks to the publisher for providing a chance by giving me with the platform to publish this book.

Contents

		Page No.
Chapter 1	Introduction	1-5
1.1	Introduction to computers	1
1.2	Introduction to computer languages	2
1.2.1	Machine level languages	2
1.2.2	Assembly level languages	2
1.2.3	High-level languages	3
1.3	Introduction to C language	3
1.3.1	History of C	4
1.3.2	Features of C	4
1.3.3	Statements in C	5
Chapter 2	C tokens & I/O functions	6-23
2.1	Introduction	6
2.2	Keywords	7
2.3	Variables	8
2.4	Identifiers	8
2.5	Constants and literals in C	9
2.6	Operators	10
2.7	Classification of C operators	12
2.7.1	Arithmetic operators	12
2.7.2	Relational operators	13
2.7.3	Logical operators	13
2.7.4	Bitwise operators	14
2.7.5	Assignment operators	16
2.7.6	Conditional operators	17
2.8	Conversion and evaluation of C expressions	17
2.8.1	Conversion of C expressions	18
2.8.2	Expression Evaluation	18
2.9	Strings	18
2.10	Special Characters	19
2.11	Formatted I/O functions	18
2.11.1	printf()	20
2.11.2	scanf()	21
2.12	Escape Sequence in C	21
Chapter 3	Algorithm & Flowchart	24-30
3.1	Introduction to Algorithm	24
3.2	Characteristics of an Algorithm	24

3.3	Advantages of Algorithms	24
3.4	Disadvantages of Algorithm	25
3.5	How to Design an Algorithm	25
3.6	Flowchart	25
3.7	Basic structure of C program	27
3.8	Enumeration Datatype	30
Chapter 4	Decision Making & Branching	32-42
4.1	Introduction	32
4.2	The if Statements	32
4.2.1	Simple if statement	32
4.2.2	The if else statement	33
4.2.3	The if else if ladder statement	36
4.2.4	nested if statement	37
4.3	The switch statements	39
4.4	The goto statement	42
4.5	The break statement	*53
4.6	The continue statement	*55
Chapter 5	Loops in C	44-57
5.1	Introduction	44
5.2	The while loop	44
5.3	The do while loop	48
5.4	The for loop	50
Chapter 6	Arrays in C	58-69
6.1	Introduction	58
6.2	Arrays in C	58
6.3	One dimensional Array	58
6.3.1	Declaring 1D array	58
6.3.2	Initialization of 1D array	59
6.3.3	Accessing 1D array elements	59
6.4	Two-dimensional array	60
6.4.1	Initialization of 2D array in C	61
6.4.2	Accessing 2D array elements	62
6.4.3	Matrix operations in C	64
Chapter 7	Strings in C	70-83
7.1	Introduction	70
7.2	Declaration of string	70
7.3	Initialization of string	70
7.3.1	Assigning a string literal	70
7.3.2	Assigning character by character	71

7.4	Accepting string as input	72
7.5	C gets() and puts() functions	73
7.5.1	Input function: gets()	73
7.5.2	Output function: puts()	73
7.6	Character oriented input/output function	74
7.6.1	C getchar() and putchar() functions	74
7.6.2	C getch() and putch() functions	75
7.6.3	C getche() functions	75
7.7	String manipulation function	76
7.7.1	strlen function	76
7.7.2	sizeof function	76
7.7.3	strcpy function	77
7.7.4	strcat function	78
7.7.5	strcmp function	79
7.7.6	strrev function	80
7.7.7	strstr function	80
7.8	scanf() and sprintf() in C	81
7.8.1	scanf function	81
7.8.2	sprintf function	81
Chapter 8	Pointers in C	84-98
8.1	Introduction	84
8.1.1	Basics to understand pointers	84
8.2	Pointers	84
8.2.1	Pointer declaration	84
8.2.2	Pointer initialization	85
8.2.3	Dangling pointer in C	86
8.2.4	Null pointer	87
8.2.5	Pointer to pointer (double pointer)	87
8.2.6	Pointer to an array	88
8.2.7	Pointer arithmetic	89
8.3	Dynamic memory allocation	93
8.3.1	The malloc() in C	93
8.3.2	The calloc() in C	95
8.3.3	The realloc() in C	96
8.3.4	The free() in C	97
8.4	Array of pointers	98
Chapter 9	Functions in C	99-111
9.1	Introduction	99
9.2	Advantages of function	99

9.3	Types of functions	99
9.4	Library functions	99
9.5	User-defined functions	100
9.5.1	Elements of user-defined functions	100
9.5.2	Types of user-defined functions	102
9.5.3	Passing parameters to functions	105
9.5.4	Function with multiple return values	107
9.5.5	Function returning pointers	108
9.6	Recursion function	109
9.6.1	Two types of recursions	111
Chapter 10	Structures and unions in C	112-121
10.1	The typedef	112
10.2	The structures in C	112
10.2.1	Declaring structure variable	114
10.2.2	The size of structure variable	114
10.2.3	Accessing the structure members	115
10.2.4	Declaring structure variable	115
10.2.5	The size of structure variable	117
10.3	The union in C	118
10.4	Similarities between structure and union	120
10.5	Difference between structure and union	120
Chapter 11	Preprocessor Directives	122-127
11.1	#include	123
11.2	Macro's(#define)	123
11.2.1	Object like macros	123
11.2.2	Function like macros	124
11.3	#undef	124
11.4	#ifdef	125
11.5	#ifndef	125
11.6	#if	125
11.7	#else	125
11.8	#error	126
11.9	#pragma	127

CHAPTER I

INTRODUCTION

1.1 Introduction to Computer

Computers are involved in all fields to make things easier. *A Computer is an electronic device that accepts the input data, stores it, and processes the data to give the information as output.* Hence, the fundamental physical components of a computer are input devices, memory, processor, and output devices. And logical component is software.

Input devices are used to provide input to the computer whereas output devices are used to get output from the computer. The memory is used to store the data. Inside the processor, computation or processing of data will be done.

The memory structures

To understand the storage of data in the memory, we can imagine computer memory as a grid structure. The computer memory consists of memory locations and each location can store one byte of data. And also, every location in the memory is assigned with a unique address. The internal structure of memory can be imagined like this below.

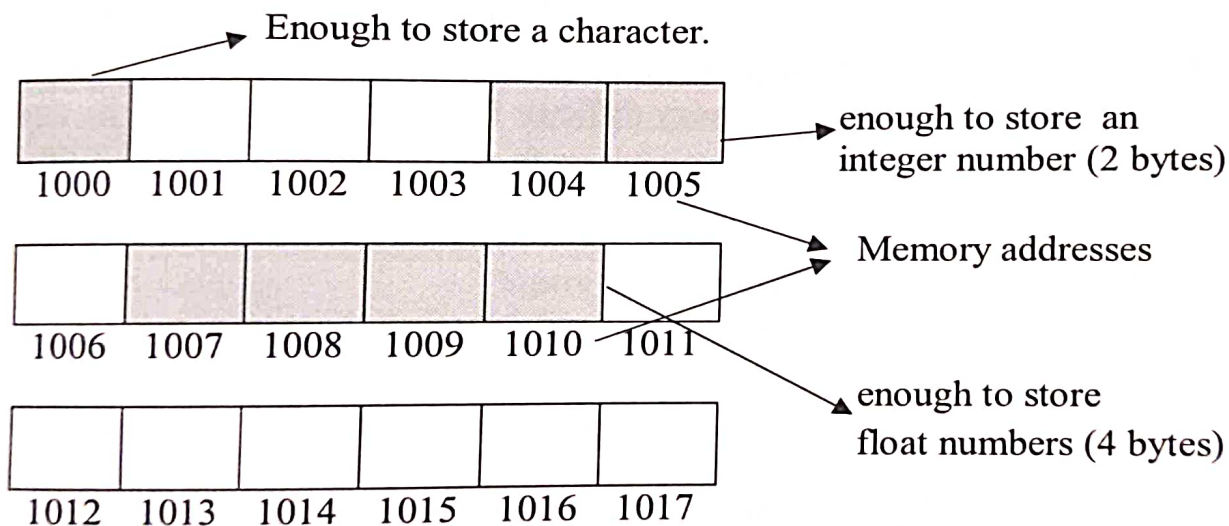


Figure 1.1: Memory structure

In the above memory, 18 memory locations are available. So, size of the above memory is 18 bytes. Here, we have taken the addresses in terms of 1000 just for the purpose of understanding the concept. The actual address format will be in hexadecimal like fffe2, fffe4 etc.