Programming and Problem Solving Through 'C' Language

Programming and Problem Solving Through 'C' Language

M3 - R4

STRICTLY AS PER NEW DOEACC SYLLABUS

RAMESH BANGIA

B.Sc., Dip. Comp. Sc. (IIT D)
Formerly: Incharge Computer Centre,
Hindustan Copper Limited, New Delhi
Ex. Managing Editor: Software Today

FIREWALL MEDIA

(An Imprint of Laxmi Publications Pvt. Ltd.)

BANGALORE ● CHENNAI ● COCHIN ● GUWAHATI ● HYDERABAD
JALANDHAR ● KOLKATA ● LUCKNOW ● MUMBAI ● RANCHI
NEW DELHI ● BOSTON, USA

Published by :

FIREWALL MEDIA

(*An Imprint of Laxmi Publications Pvt. Ltd.*)
113, Golden House, Daryaganj,
New Delhi-110002

Phone : 011-43 53 25 00
Fax : 011-43 53 25 28

www.laxmipublications.com
info@laxmipublications.com

Price : ₹ **295.00** *Only.* *New Edition*

OFFICES

✆ **Bangalore**	080-26 75 69 30	✆ **Chennai**	044-24 34 47 26
✆ **Cochin**	0484-237 70 04, 405 13 03	✆ **Guwahati**	0361-251 36 69, 251 38 81
✆ **Hyderabad**	040-24 65 23 33	✆ **Jalandhar**	0181-222 12 72
✆ **Kolkata**	033-22 27 43 84	✆ **Lucknow**	0522-220 99 16
✆ **Mumbai**	022-24 91 54 15, 24 92 78 69	✆ **Ranchi**	0651-221 47 64

FPR-3108-295-PROG PROB SOL TH C LANG-BAN **C—**
Typeset at : Kalyani Computer Services, Delhi. *Printed at* :

Contents

Preface

The new syllabus of DOE's various levels of courses, popularly called O-level, A-level, B-level and C-level is out. Keeping in mind the various developments taking place around the world in the field of computers, these syllabus include most of them. While keeping in mind the level of students studying the various levels, the degree of difficulties of the syllabus have been very well designed. I hope that after the new syllabus the courses would be as popular as they were before.

There had been a long struggle between Training Institutes and Government Organizations, since these organizations would not recognize the degree/diploma given by them. Students passing out from these institutions were not given jobs based upon the said diplomas/degrees. Ultimately, the Government formed a society under the Department of Electronics, called DOEACC, who conducts tests and gives grades to the students irrespective of the institutes where they have studied. It is also possible to give the test without having any proper institutional studies. This is more or less on the terms of SAT and GRE conducted by USA. The only difference is that this is not online. There are fixed dates for the exams.

The course for this has been designed for students to appear in the exams at various levels. These levels are O-level, A-level, B-level, C-level. The most popular among these is the O-level, since it is the entry level. Once you have cleared the O-level, you are eligible for a government job. Each course has been divided into various modules. Various books are available for various modules. This book is also made for you to prepare for the O-level course.

This book has been written totally in accordance with the syllabus provided. Each chapter has been embedded with the screen shots for you to know what to expect on the screen, while running the application. Each chapter contains examples which can be attempted to supplement the reading. At the end of the chapter, there is a thorough revision of the text read. Not to mention of the questions provided in the beginning, both in the form of objective and subjective. The book also contains Sample papers which are based on the samples provided by DOEACC society. These would help you in preparing for the exams.

–Author

SYLLABUS

M3-R4: PROGRAMMING AND PROBLEM SOLVING THROUGH 'C' LANGUAGE

The objective of this course are to make the student understand programmng language, programming, concepts of loops, reading a set of data, stepwise refinement, Functon, Control strucutres, Arrays. After completion of this course the student is expected to analyze the real life problem and write a program in 'C' language to solve the problem. The main emphases of the course will on problem solving aspect, i,e., developing proper algorithms.

At the end of the course the students will be able to:

- Develop efficient algorithms for solving a problem.
- Use the various constructs of a programming language viz., conditional, iteration and recursion.
- Implement the algorithm in 'C' language.
- Use simple data structures like arrays, stacks, and linked list in solving problems.
- Handling File in 'C'.

Outline of Course

S.No.	*Topic*	*Minimum No. of Hours*
1.	Introduction to Programming	04
2.	Algorithm for Problem Solving	10
3.	Introduction to 'C' Language	04
4.	Conditional Staements and Loops	07
5.	Arrays	06
6.	Functions	06
7.	Storage Classes	03
8.	Structures and Unions	06
9.	Pointers	06
10.	Self Referential Structures and Linked Lists	04
11.	File Processing	04

Lectures	=	60
Practicals /Tutorials	=	60
Total	=	120

DETAILED SYLLABUS

1. Introduction to Programming **04 Hrs.**

The Basic Model of Computation, Algorithms, Flow-Charts, Programming Languages, Compilation, Linking and Loading, Testing and Debugging, Documentation.

2. Algorithms for Problem Solving **10 Hrs.**

Exchanging values of two variables, Summation of a set of numbers, Decimal Base to Binary Base Conversion, Reversing digits of an integer, GOD (Greatest Common Division) of two numbers, Test whether a number is prime, Organize numbers in ascending order, Find square root of a number, Factorial computation, Fibonacci sequence, Evaluate 'sin x' as sum of a series, Reverse order of elements of an array, Find largest number in an array, Print elements of upper triangular matrix, multiplication of two matrices, Evaluate a Polynomial.

3. Introduction to 'C' Language **04 hrs.**

Character set, Variables and identifiers, Built-in Data Types, Variable Definition, Arithmetic operators, and expressions, Constants and Literals, Simple assignment statement, Basic input/output statement, Simple 'C' programs.

4. Conditional Statements and Loops **07 Hrs.**

Decision making within a program, Conditions, Relational Operators, Logical Connectives, If statement, If-else statement, Loops, While loop, Do while, For loop, Nested loops, Infinite loops, Switch statement, Structured programming.

5. Arrays **06 Hrs.**

One dimensional arrays, Array manipulation; Searching, Insertion, Deletion of an element from an array; Finding the largest/smallest element in an array; Two dimensional arrays; Addition/Multiplication of two matrices, Transpose of a square matrix; Null terminated strings as array of characters, Standard library string functions.

6. Functions **06 Hrs.**

Top-down approach of problem solving, Modular programming and functions, Standard Library of C functions, Prototype of a function; Format parameter list, Return Type, Function call, Block structure, Passing arguments to a Function; call by reference, call by value, Recursive Functions, arrays as function arguments.

7. Storage Classes **03 Hrs.**

Scope and extent, Storage Classes in a single source file; auto, extern and static, register, Storage Classes in a multiple source files; extern and static.

8. Structures and Unions **06 Hrs.**

Structure variables, initialization, structure assignment, nested structure, structures and functions, structures and arrays; arrays of structures, structures containing arrays, unions.

9. Pointers **08 Hrs.**

Address operators, pointer type declaration, pointer assignment, pointer initialization, pointer arithmetic, functions and pointers, Arrays and Pointers, pointer arrays, pointers and structures, dynamic memory allocation.

10. Self Referential Stuctures and Linked Lists **04 Hrs.**

Creation of a singly connected linked list, Traversing a linked list, Insertion into a linked list, Deletion from a linked list.

11. File Processing **04 Hrs.**

Concept of Files, File opening in various modes and closing of a file, Reading from a file, Writing onto a file.

Chapter

1 Introduction to Programming

1.1 THE BASIC MODEL OF COMPUTATION

A good programmer is an asset to the company. But, nobody is born to write programs. A good programmer is not born but is developed by his own actions. These actions make him a good programmer. There is a methology which he has to follow to become a good programmer. In this chapter I will discuss this methology, and how it is used while writing programs irrespective of the language he is using.

Programming is a team effort. What can a good programmer do if the inputs to him are given wrong. He has to coordinate between the various users and data providers to cater to their needs and write a good program which would produce the results the user is looking forward to. A manager, for example, is not concerned how the problem is solved or how the data is obtained. He is only interested in results which the program is supposed to provide.

For writing any program the programmer has to keep lots of things in mind. Few of them have been mentioned above, the others including problem definition, program design, coding, debugging, testing, documentation, maintenance and redesign. Let us start with the general concepts used for writing any program.

1.1.1 General Concepts

Program writing is definitely a technique. One has to master it by experience. One should always start with writing small and simple programs and then graduate to complex and complicated programs. Whatever he may write, it should be written in a manner which is understood by others. God forbid if he is not around the others should be able to run and modify it. Here the documentation parts becomes very necessary. I will talk about it little later but first let us see what a good program should have.

A good program should have the following characteristics:

1. Readability
2. Efficiency
3. Reliability
4. Meaningfulness
5. Portability
6. User-friendly.

Let us read about them in little details.

1.1.1.1 Readability

As mentioned above the program should be such that it can be read and understood by others. This makes it easy to modify by others.

1.1.1.2 Efficiency

While writing the program the programmer must take care of the following:

1. Processing

He should make sure that the program takes the minimum processing time.

2. Memory

Mimimum utalization of memory and hard disk space should be used.

3. Time

The programmer should be able to produce the program in time frame given.

1.1.1.3 Realiability

It goes beyond saying that the program should be reliable and must be in a position to produce good and accurate results. It should take care that the various checks are provided to overcome wrong data and machine malfunctions.

1.1.1.4 Meaningfulness

The program should give the meaningful outputs.

1.1.1.5 Portability

With the advent of PCs the compatability of one computer with another has been taken care of. But, still there is a possibility that the program may be required to run on a machine which is alien to the program. In this case the program should be such that it needs minimum efforts to be modified to suit the new environments.

1.1.1.6 User-friendly

A program should be user-friendly.

1.2 PROBLEM SOLVING

It is impossible to solve a problem by using a computer, without a clear understanding and identification of the problem. Inadequate recognition of a problem is the key element responsible for poor performance of computers. This step is generally difficult and the programmer should invest a significant portion of his time in problem identification. If he does not spend enough time at this state, he may find that his well-written program fails to solve the real problem.

1.2.1 Understanding of the Problem

Depending on the complexity of the problem, the result of a careful analysis may be simply a testing of the factors with which the program must deal, or it may be a formal written statement of the problem, supported by a description of all the considerations necessary to solve the problem. This step is the process of becoming familiar with the problem that will be solved with a computer program. It starts when the programmer is assigned the task.

This step includes the reviewing of the design document that was prepared for the program, as well as any system-wide information that would be helpful. The process ends when all the programmer's questions have been resolved and the requirements of the program are understood.

Problem definition may, thus, include many factors. Input/Output, time constraints, processing requirements, accuracy, memory limitations, error handling and interfaces with other programs or tables must all be considered.

1.2.2 Identifying Minimum Number of Inputs required for Output

In any problem soloving exercise, the computer center receives information from various departments and sends the processed output to various other departments. In most of the cases, the computer center does not have control over the input of the data. Each department which sends the data for processing has to ensure that the data sent is free from errors. But can the computer center make its own checks, yes, they can provide various checks in the data. For example, they can make sure that the number of days worked in never beyond 31, the age of the employee is never less than 18, the basic salary is not less than DA, etc. But still if any error occurs, it is the fault of input and not the processing.

Human factor is also important here. Since the date and time of receiving data is fixed, the department sending the data is under pressure for the same. This results in wrong filling up of the data sheet. This results in wrong results. These data sheets are the results of the various sittings which the computer center has had with both the users and input providers.

1.2.3 Step by Step Solution for the Problem

An often used rule is that a programmer working on a large software project, will write two to ten fully corrected lines of program per day. Since the writing of even ten lines hardly takes just a few minutes, coding is clearly not the most time-consuming stage of software development. So what he does he chalks out the various steps in which the program has to be written. The solution of the problem lies in good steps.

A program writing can be divided into several stages:

1. Program Definition
2. Program Design
3. Coding
4. Debugging
5. Testing
6. Documentation
7. Maintenance

Some of them have been discussed earlier.

Once these steps have been marked down the next step becomes easier.

1.2.4 Breaking Down Solution into Simple Steps

This step is considered complete when a clean compile walkthrough has been held and the program coding has been approved. These steps would not only result in good program but would give satisfactory results.

Stylistic Guidelines: This section presents stylistic guidelines on programming style that will be useful, regardless of the particular language or processor involved.

Programming And Problem Solving Through C Language M3-R4 By Ramesh Bangia

Publisher : Laxmi Publications ISBN : 9789380298900 Author : Ramesh Bangia