PROGRAMMING FUNDAMENTALS USING 'C'

COURSE OVERVIEW

The computing world has undergone a revolution since the publication of The C Programming Language in 1978. Big computers are much bigger, and personal computers have capabilities that rival mainframes of a decade ago. During this time, C has changed too, although only modestly, and it has spread far beyond its origins as the language of the UNIX operating system.

The growing popularity of C, the changes in the language over the years, and the creation of compilers by groups not involved in its design, combined to demonstrate a need for a more precise and more contemporary definition of the language than the first edition of this book provided. In 1983, the American National Standards Institute (ANSI) established a committee whose goal was to produce "an unambiguous and machine-independent definition of the language C", while still retaining its spirit. The result is the ANSI standard for C.

It provides a new form of function declaration that permits cross-checking of definition with use. It specifies a standard library, with an extensive set of functions for performing input and output, memory management, string manipulation, and similar tasks.

It makes precise the behavior of features that were not spelled out in the original definition, and at the same time states explicitly which aspects of the language remain machine-dependent.

Although we have noted the places where the language has evolved, we have chosen to write exclusively in the new form. For the most part, this makes no significant difference; the most visible change is the new form of function declaration and definition. Modern compilers already support most features of the standard.

C is not a big language, and it is not well served by a big book. We have improved the exposition of critical features, such as pointers, that are central to C programming. We have refined the original examples, and have added new examples in several chapters. For instance, the treatment of complicated declarations is augmented by programs that convert declarations into words and vice versa. As before, all examples have been tested directly, which is in machine-readable form.

As we said in the preface to the C "wears well as one's experience with it grows". With a decade more experience, we still feel that way. We hope that this book will help you learn C and use it well.

Programming Fundamentals Using C

BLOCK 1

UNIT 1:

Concept of Data Storage within a Computer Program, Computer Memory, Concept of Variables, Constants and Preprocessor Directive Statements

UNIT 2:

Elements of Language: Expressions, Statements, Operators: Binary operators, Relational operators

UNIT 3:

Branching Statements, Evaluating Relational Expressions, Precedence of Relational operators, Logical operators

UNIT 4:

Controlling Program execution, **While** statement, **Do while** Loop, Nested Loop

BLOCK 2

UNIT 5:

Ending Loops Early, The Break statement, The Continue statement, **goto** statement, Switch Statements

UNIT 6:

Functions, How a function works, Function prototype, Recursive function

UNIT 7:

Introduction to arrays, One-dimensional & Multi dimensional arrays, Naming & Declaring Arrays

UNIT 8:

Types of I/O, Console I/O function, Escape sequences

BLOCK 3

UNIT 9:

Formatted output conversion specifiers, Character input and character output, Type conversion, Stream I/O

UNIT 10:

Scope of Variable, Global & Local Variables, Scope of function parameter, extern static and register variable

UNIT 11:

Input output redirection, Command line argument

UNIT 12:

Introduction to structure and Unions, and defining a structure with examples, array of structures, initializing structures

BLOCK 4

UNIT 13:

Introduction to Typedef and macros, Details of Union and Programs related to Unions

UNIT 14:

Introduction to Bits, Bit wise operators, complement operators, Dynamic Memory allocation, malloc, realloc, calloc functions

UNIT 15:

Verification and validation, testing process, test strategies

UNIT 16:

Error handling functions, Types of errors

BOOKS AND REFERENCES:

1. Programming in ANSI C ,E. Balagurusamy
2. Programming in C, Kris A. Jamsa
3. Let us C , Yashwant Kanetkar
4. Programming with C , Shaum's Publication