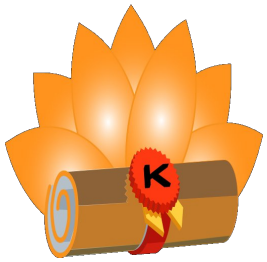
www.kamaldnp.com



KAMALDNP

kamaldnp.com

PDF created by Raj Kamal using AI

Computer Programming Using C — Complete Syllabus

BTEUP Diploma in Information Technology • 3rd Semester • Session 2026–27

Subject 3.7 (Practicum) • Total 100 Marks • 3 Credits • L 1 – P 4

About this document: Computer Programming Using C is subject 3.7 of the 3rd semester of the Diploma in Information Technology. It is a practicum (1 lecture and 4 practical periods a week) with NO theory paper: assessment is 60 marks internal practical and 40 marks external practical (3 hours). This document gives the complete unit-wise syllabus, the examination scheme, the practical exercises, the course outcomes and the reference books from the official curriculum. The syllabus prints no separate titles for the five units, only numbers; the headings below use the topic lead-in printed at the start of each unit. The syllabus prints “tertiary operators” in Unit 2; the correct term is ternary operators.

Examination Scheme

Code	Paper	Type	L – P	Credits	Internal	External	Total
3.7	Computer Programming Using C	Practicum	1 – 4	3	Practical 60	Practical 40 (3 hrs)	100
Computer Programming Using C Total				**3**	**60**	**40**	**100**

What This Subject Aims To Do

To enable student, develop structured solutions to problems and implementing them using computers. This involves two parts: (i) formulating a solution for a given problem as a well-defined sequence of actions, and (ii) expressing solution in a machine-readable form or a

www.kamaldnp.com

programming language. For the second part, we will learn the common units of programming languages. The first part can only be learned through the repeated practice of solving problems.

Unit-Wise Complete Topic List

Unit	Topic	Periods	Marks (%)
1	Introduction to Problem Solving	2	20
2	Operators and Expression	3	20
3	Conditional and Looping Statements	3	20
4	Arrays and String	3	20
5	Function and Pointer	3	20

Total: 14 periods, 100%. The percentages are the suggested distribution of marks in the official syllabus, out of the 40-mark external practical examination. All five units carry the same 20%, so no unit can be skipped.

Unit 1 — Introduction to Problem Solving (2 Periods, 20%)

Topics: Introduction to problem solving: computational way of thinking, flowchart and algorithm, variable, constant and representation, data types, keywords, identifier.

Practical exercise: Installation of C compiler. Familiarisation with the programming environment and options/menu of the installed compiler. Draw a flowchart using MS Word (or any other editor or online tools like Draw.io etc.).

In simple words: Thinking before coding — how to break a problem into steps (algorithm), draw those steps (flowchart), and the basic building blocks of C: variables, constants, data types, keywords and identifiers.

Unit 2 — Operators and Expression (3 Periods, 20%)

Topics: Operators and expression: arithmetic, relational, logical and bitwise operators, input, output, expressions, data type casting.

Practical exercise: Programs using I/O statements and various operators: print your name; perform various arithmetic and logical operations; swap two numbers with and without using a third variable. Programs using expression evaluation and precedence: using unary and ternary operators (the syllabus prints “tertiary”); using post and pre-increment operators.

In simple words: Doing things with values — the operators that calculate, compare and combine, reading input and printing output, and changing a value from one type to another.

www.kamaldnp.com

Unit 3 — Conditional and Looping Statements (3 Periods, 20%)

Topics: Conditional and looping statements: if, if-else, if-else ladder, switch-case, break and continue. Loops: for, while and do-while, and nested loops.

Practical exercise: Programs using decision-making and branching statements: check whether a given number is odd or even; check whether a given number is prime or not; find the largest among three given numbers. Programs using loop statements: find the factorial of a given number; print the table of a given number; print all prime numbers from 1 to 100; print the Fibonacci series up to a given number n; print different star patterns.

In simple words: Making the program decide and repeat — if-else and switch for choices, for, while and do-while for repetition, and the classic practice programs: prime numbers, factorials, tables and patterns.

Unit 4 — Arrays and String (3 Periods, 20%)

Topics: Arrays and strings: introduction to arrays, array declaration, length of array, manipulating array elements, single and multidimensional array, arrays of characters, introduction to strings, string declaration and definition, string-related functions: strlen, strcpy, strcmp.

Practical exercise: Programs to demonstrate applications of n-dimensional arrays: print a 3 x 3 matrix; print the transpose of a 3 x 3 matrix; add two matrices. Programs to demonstrate use of string manipulation functions: find the length of a given string; concatenate two strings; check whether a given string/number is palindrome or not.

In simple words: Holding many values under one name — arrays for lists and tables (matrices), strings for text, and the library functions that measure, copy and compare strings.

KAMALDNP

www.kamaldnp.com

Unit 5 — Function and Pointer (3 Periods, 20%)

Topics: Function and pointer: introduction to functions, global and local variables, function declaration, function call and return, types of functions, standard functions, introduction to pointer, parameters and parameter passing, call-by-value/reference, recursive function, I/O and formatting.

Practical exercise: Programs to demonstrate parameter passing: create a user-defined function to add two numbers; implement call by value; implement call by reference. Programs to demonstrate recursion: print the factorial of a given number using recursion. Programs to demonstrate use of pointers: print the address of a variable using a pointer; swap the contents of two variables using a pointer.

In simple words: Breaking a program into reusable pieces — functions, how values are passed to them, recursion (a function calling itself), and pointers: variables that hold the address of another variable.

Course Outcomes

- CO1: Demonstrate knowledge on C Programming concepts.
- CO2: Develop simple programs in C using basic constructs.
- CO3: Develop C programs using conditional and looping concepts.
- CO4: Understand the concept of multi-dimensional array and strings.
- CO5: Design modular C programs with reusable functions and pointers to improve code readability and maintainability.

KAMALDNP • kamaldnp.com • support@kamaldnp.com
Source: BTEUP IRDT Kanpur NEP-2020 curriculum for Diploma in Information Technology, effective session 2026-27 (urise.up.gov.in). Topic lists follow the official syllabus; short explanations are the author's own plain-English summaries, not syllabus text. Please confirm with your own polytechnic for any local changes.