UG CA S3 01 (DSC)

FOUR YEAR UNDERGRADUATE PROGRAMME

(CBCS in Accordance with NEP 2020)

SEMESTER - III

PROGRAMMING IN C

BACHELOR OF COMPUTER APPLICATION

Hiranya Chandra Bhuyan School of Science and Technology
KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

Subject Experts

Prof. Shikhar Kumar Sarma, Department of Information Technology, Gauhati University
Prof. Hiren Kumar Deva Sarma, Department of Information Technology, Gauhati University
Dr. Sanjib Kr. Kalita, Department of Computer Science, Gauhati University

Course Co-ordinator : **Dr. Ridip Dev Choudhury,** KKHSOU
Dr. Tapashi Kashyap Das, KKHSOU
Dr. Sruti Sruba Bharali, KKHSOU

SLM Preparation Team

UNITS	AUTHORS
1, 4, 6, 11, 13	**Dr. Tapashi Kashyap Das**, KKHSOU
2, 3, 7	**Arabinda Saikia**, KKHSOU
5, 12	**Dr. Sanjib Kr. Kalita**, Gauhati University (Former Faculty, KKHSOU)
9, 10	**Dr. Sruti Sruba Bharali**, KKHSOU
8, 14	**Binod Deka**, KKHSOU

Editors

Content **Dr. Tapashi Kashyap Das**, KKHSOU (Unit 2, 3, 5, 7, 9, 10, 12)
Dr. Sruti Sruba Bharali, KKHSOU (Unit 1, 4, 6, 8, 11, 13, 14)

Language: **Prof. (Retd.) Robin Goswami,** Cotton College

Structure & Format : **Dr.Tapashi Kashyap Dase**, Assistant Professor, KKHSOU

Print: September:2024 ISBN-978-93-48100-67-2

This Self Learning Material (SLM) of the Krishna Kanta Handiqui State Open University is made available under a Creative Commons Attribution-Non Commercial-Share Alike 4.0 License (international) http://creativecommons.org/licenses/by-nc-sa/4.0/

Printed and published by Registrar on behalf of the Krishna Kanta Handiqui State Open University.

Headquarters: Patgaon, Rani Gate, Guwahati - 781 017
City Campus: NH 37, Resham Nagar, Khanapara, Guwahati - 781 022;

www.kkhsou.in

CONTENTS

		PAGE NO
UNIT 1 :	**INTRODUCTORY CONCEPTS** Basic definition of Pseudo Code, algorithm, flowchart, program.	**7-25**
UNIT 2 :	**ELEMETS OF C PROGRAMMING** Characters used in C, Identifiers, Keywords, Tokens, Constants, Variables.	**26-31**
UNIT 3 :	**VARIABLES AND DATA TYPES** Integer, character floating point and string; Initialization of variable during declarations; Symbolic Constants.	**32-39**
UNIT 4 :	**OPERATORS AND EXPRESSIONS** Expression in C, Different types of operators: Arithmetic, Relational and Logical, Assignment, Conditional, Increment and decrement, Bitwise, Comma and other operator (sizeof, period etc). Precedence and associatively of operators, type casting	**40-67**
UNIT 5 :	**PREPROCESSOR DIRECTIVES AND I/O FUNCTIONS** Header Files (stdio, conio), Formatted Input / Output Functions (scanf, printf), Escape Sequences, Character Input/Output Functions (getch, getchar, putchar, gets, puts, getche, clrscr).	**68-80**
UNIT 6 :	**CONDITIONAL STATEMENTS** Conditional Statement- if, if- else, nested if-else, switch-case; break, continue, goto	**81-98**
UNIT 7 :	**LOOP CONTROL STRUCTURES** Concept of Loops, Types of loop: while, do-while, for; nested loops.	**99-112**
UNIT 8 :	**STORAGE CLASS** Automatic, External, Static, Register, Scope and lifetime of variables, Macro, Preprocessor directives.	**113-127**
UNIT 9 :	**ARRAYS** Array, Array Declaration, 1-Dimensional array, 2-Dimensional array.	**128-143**

UNIT 10 : STRINGS **144-160**

String, String Handling Functions: strlen (), strcmp (), strcpy (), strrev (), strcat (), etc.

UNIT 11 : FUNCTIONS **161-185**

Function, Function declaration, Function definition, Function call, Formal and Actual parameter, Recursive function.

UNIT 12 : POINTERS **186-199**

Declaring Pointer Variable; Pointer Arithmetic, Pointers and One-Dimensional Arrays, Pointers and Character Arrays, Pointer declaration, Passing Pointer to a Functions as Argument, Dynamic Memory Allocation: malloc (), calloc (), realloc (), free ().

UNIT 13 : STRUCTURE AND UNION **200-222**

Structure Declarations, Definitions, Defining your typedef, Array of Structure, Pointer to Structure. Union Declaration, Definition, Declaration, Uses.

UNIT 14 : FILE HANDLING **223-240**

Concept of File, File Pointer, File Opening in various modes, Closing a file, Reading and writing on files, Formatted Input / Output, fseek (), ftell (), rewind ().

COURSE INTRODUCTION: PROGRAMMING IN C

Programming languages serve as the medium through which humans communicate instructions to computers, which cannot understand human languages. Among these, C stands out as one of the most widely-used programming languages, with applications ranging from system software to application software. Developed in the early 1970s by *Dennis Ritchie* at Bell Laboratories, USA, C has become foundational in the field of computer science.

The course starts with an introduction to programming. You will learn the basics of programming languages, how to write algorithms, create step-by-step procedures for solving problems, and use flowcharts to show these processes visually. It then delves into basic constructs, including various data types that can be manipulated within a program, operators and expressions for performing operations on data, and preprocessor directives and macros that provide instructions before compilation. Control statements like if and if-else for conditional execution, switch for multi-way branching, and break and continue for loop control mechanisms are also explored. Iteration is discussed through loop concepts, including for, while, and do-while loops for repeating sections of code. The course then advances to modular blocks of code designed for specific tasks, such as functions, and covers collections of data items of the same type, like arrays, as well as arrays of characters for handling textual data, known as strings. Pointers, which store memory addresses and facilitate dynamic memory allocation and manipulation, are also examined. Storage and data structures are introduced, covering storage classes that define the scope, visibility, and lifetime of variables, custom data types for grouping different data types together, known as structures, and unions, which are similar to structures but share a memory space for all members. Finally, the course addresses file handling, including essential file operations for reading from and writing to files, ensuring data persistence.

By the end of this course, learners will have a comprehensive understanding of the C programming language, enabling them to develop efficient and effective programs. Through rigorous practice, learners will develop their programming abilities, gaining a strong foundation that will facilitate learning other programming languages.

While going through a unit, you will notice some boxes, containing such information as would help you to understand some of the difficult, unfamiliar terms. You will find an item called “ACTIVITY’ where you will apply your knowledge and thoughts in some questions pertaining to the relevant topic. Again, we have included some relevant concepts in “LET US KNOW” along with the text. And, at the end of each seection, you will get “CHECK YOUR PROGRESS” questions. These have been designed to self-check your progress of study. It will be better if you solve the problems put in these boxes immediately after you go through the sections of the units and then match your answers with “ANSWERS TO CHECK YOUR PROGRESS” given at the end of each unit.