Semester: II	
Programme : B.Sc. Computer Science (Hons)	
Course : PROBLEM SOLVING TECHNIQUES USING C	
Paper code: B1CS230212T / B1CS230212P	**Credits: 4**
Hours/week : Theory: 3 / Practical 2	
Category: Core/MDC/SEC/VAC : Minor	
Theory / Practical / Composite : Composite	
No of Modules : 1	

Course Overview: Problem Solving Techniques using C is a 4-credit composite course designed to introduce B.Sc. Computer Science students to fundamental programming concepts using the C language within the procedural paradigm. The curriculum covers programming language generations, C syntax fundamentals, data types, operators, control structures, functions, arrays, pointers, structures, unions, storage classes, and macros. Through integrated theory and practical sessions, students develop skills to build interactive and modular programs while understanding memory management techniques—establishing a strong foundation for advanced programming studies.

Course Outcome:

1. **Recall and explain** fundamental programming concepts and C language basics, understand core elements of C such as program structure, character set, keywords, identifiers, variables, constants, and standard library overview.
2. **Apply** basic C constructs to develop interactive programs, use data types, operators, expressions, standard I/O functions, decision-making statements (if-else, switch-case), and iterative constructs (for, while, do-while) to write simple yet functional procedural programs.
3. **Analyze** modular programming through functions and storage classes, interpret and implement user-defined functions—including recursive functions—and manage variable scope and lifetime using appropriate storage classes (auto, static, extern, register).
4. **Design** structured data handling and memory management techniques, assess and effectively use arrays, pointers, structures, and unions to model complex data and manage memory dynamically, understanding the relationship between pointers and arrays or functions.
5. **Evaluate** memory management techniques using pointer operations, and demonstrate the implementation of call-by-value and call-by-address mechanisms in function parameter passing.
6. **Create** and integrate modular design, memory management, structured data and preprocessing to build complete, robust, and application-oriented C programs for real-life problem-solving in Information Technology Enabled Services contexts.

Prerequisites:

Basic computer literacy, fundamental mathematics, logical and analytical thinking skills, familiarity with text editors and command line interfaces.

SYLLABUS

UNIT/Module	CONTENT	HOURS	CO Mapping	COGNITIVE LEVEL
I.	Generations of Programming Languages: Machine Language, Assembly Language, Procedural Language, Object Oriented Language. Introduction to C Programming Language: Features and Structure of a C Program, Character Set, Identifiers and Keywords, Variables and Constants, Brief Idea about C Library.	5	CO1	K1, K2 (Remember/Understand)

II.	Data Types in C: Primitive, User-Defined, Enumerated, Type Casting, Declaration. Operators in C: Different Types, Precedence and Associativity, Expressions using Operators.	5	CO2	K3 (Apply)
III.	Input-Output Operations: Standard Functions with Escape Sequences and Format Specifiers. Decision Making Statement: if-else, switch-case, Ternary Operator.	5	CO2	K3 (Apply)
IV.	Iterative Statements: for, while and do-while with control statements like break and continue.	5	CO2	K3(Apply)
V.	Functions: Declaration, Calling and Definition, Idea about Recursive Function. Storage Classes.	6	CO3	K4 (Analyze)
VI.	Arrays: 1-D and 2-D Declaration and Usage; Strings; Passing Arrays to Functions. Pointers: Declaration and Usage; Call-by-Value vs. Call-by-Address; Passing Pointers to Functions.	6	CO4	K5 (Evaluate)
VII.	Structures and Unions: Declaration and Usage.	4	CO4	K5 (Evaluate)
VIII.	Macros: Types, Declaration and Usage.	3	CO5, CO6	K6 (Create)

Text Books		
1. The C Programming Language, Kernighan and Ritchie, PHI Publications		
2. Programming with C, Gottfried, TMH. Publications		
3. Programming in C, Dey and Ghosh, Oxford Publications		
4. Programming in ANSI C, Balaguruswamy, McGraw Hill		
Suggested readings		
Let Us C, Yashavant Kanetkar, BPB Publications		
Web Resources		
1. NPTEL course on Introduction to Programming in C by Dr. Satyadev Nandakumar, IIT Kanpur; course link: https://youtu.be/XTiIiI-LOY8 2. SWAYAM Course on Art of C Programming by Dr. Lajish V.L, Department of Computer Science, University of Calicut.		
Evaluation	Theory CIA: 12 Attendance: 3 Semester Exam: 45	Practical CA: 38 Attendance: 2
Paper Structure for Theory Semester Exam Module : Answer 3 out of 5 of 15 marks each		

Course outcomes (COs) and Cognitive Level Mapping

COs	CO Description	Cognitive levels
CO1	Recall and explain generations of programming languages and fundamental C programming concepts including features, program structure, character set, identifiers, keywords, variables, constants, and C library overview.	K1-K2 (Remember/Understand)
CO2	Apply C language constructs including data types, operators, expressions, standard I/O operations, decision-making statements, and iterative control structures to develop basic interactive programs	K3 (Apply)
CO3	Analyze modular programming principles through function declaration, calling, definition, and recursive function implementation along with storage class mechanisms for variable scope management.	K4 (Analyze)
CO4	Design composite data representations using arrays (1-D/2-D), strings, structures, and unions to model real-world problem scenarios.	K6 (Create)
CO5	Evaluate memory management techniques through pointer operations and implement call-by-value versus call-by-address mechanisms in function parameter passing.	K5 (Evaluate)
CO6	Create reusable code components using macros and integrate various C programming constructs to develop complete procedural solutions for real-world problem scenarios.	K6 (Create)