

Core 2 For the students admitted from A.Y. 2023-2024& onwards		
Offering Department: Computer Application		Offered to: Master of Computer Application
Semester – I		
Course Code	Course Title	Course Credit and Hours
23MCACC102	Core 2: Problem Solving Methodologies using C Language	3 Credits - 3 hrs/wk

Course Description:

Introduction to C Programming is a course that provides students with a foundation in programming using the C programming language. The course is designed to equip students with the skills and knowledge required to write efficient and effective code and design algorithms in C. Throughout the course, students will learn about programming concepts such as variables, data types, control structures, arrays, functions, and pointers. They will also gain an understanding of software development, debugging, and testing. At the end of the course, students will have gained a strong foundation in C programming, including an understanding of the syntax and semantics of the language. They will be able to write programs using variables, data types, control structures, arrays, functions, and pointers. Additionally, they will be able to design and implement algorithms in C, debug and test programs, and apply software development principles to build software solutions using C.

Course Purpose:

C programming helps students to develop their problem-solving skills by enabling them to break down complex problems into smaller, more manageable pieces. C programming also teaches students about computer memory management, including concepts such as pointers, arrays, and dynamic memory allocation. Writing efficient and optimized code is another key aspect of C programming, which is essential for developing high-performance software. Moreover, C programming provides a strong foundation for students to learn more advanced programming languages, such as Java, Android and Python. Therefore, teaching C programming in university syllabus is an important part of computer science education that helps students to develop their programming skills and prepare them for careers in software development and related fields.

Course Outcomes: Upon completion of this course, the learners will be able to

CO No.	CO Statement	Bloom's Taxonomy Level (K ₁ to K ₆)
CO ₁	Apply control structures such as loops and decision-making constructs to solve programming problems that involve file handling	K2
CO ₂	Utilize C language features to develop efficient and scalable programs that address regular problems involving file handling	K5
CO ₃	Debug and troubleshoot C code using appropriate tools and techniques, including debugging file I/O errors	K4
CO ₄	Write simple C programs that utilize variables, data types, and basic input/output operations, including file I/O	K1
CO ₅	Design and implement complex algorithms using functions, arrays, pointers, and file handling operations	K3

Course Content	Hours
Unit-I: Introduction to C concepts	9 hrs

Course Content	Hours
<ul style="list-style-type: none"> ▪ Modular Programming ▪ Structured Programming ▪ Algorithms and Flowcharts ▪ Character set, C tokens, Identifiers, Keywords ▪ Constants, variables, data types, declaration of variables ▪ Declaring a variable as constant, declaring a variable as volatile ▪ Operators in C ▪ Type Conversions 	
Unit-II: Managing I/O Operations, Control Structures	9 hrs
<ul style="list-style-type: none"> ▪ The scanf() & printf() functions for input and output operations ▪ A character, writing a character, (the getchar() & putchar() functions) ▪ The address operator (&), formatted input and output using format specifiers, Writing simple complete C programs ▪ Control Statements: <ul style="list-style-type: none"> • Decision making with if statement, simple if statement, the if else statement, nesting of if else statements, the else If ladder, the switch statement • the ":" operator, the goto statement, the break statement ▪ Loop Control Structures: <ul style="list-style-type: none"> • the break statement, the do While statement, the for statement, nested loops, jumps in loops, the continue statement 	
Unit-III: Functions , Arrays and String handling	9 hrs
<ul style="list-style-type: none"> ▪ Functions: <ul style="list-style-type: none"> • Function Definition, Prototyping, Types of functions, passing arguments to functions, Nested Functions, Recursive functions ▪ Arrays: <ul style="list-style-type: none"> • Declaring and Initializing, One Dimensional Arrays, Two Dimensional Arrays, Multi-Dimensional Arrays, Passing arrays to functions ▪ Strings: <ul style="list-style-type: none"> • Declaring and Initializing strings, Operations on strings, Arrays of strings, Passing strings to functions 	
Unit-IV: Structures, Unions and Pointer	9 hrs
<ul style="list-style-type: none"> ▪ Structures and Unions <ul style="list-style-type: none"> • Defining a structure, Declaring structure variables, accessing structure members, Structure initialization, Copying and comparing structure variable operations on individual members, Array of structures, structures within structures, Structures and functions, Size of structures, Bit fields ▪ Unions ▪ Pointers: <ul style="list-style-type: none"> • Understanding pointers, Accessing the address space of a variable, Declaring and Initialization Pointer Variables, Accessing a Variable through its Pointer • Chain of Pointers, Pointer Expressions, Pointers and Arrays, Pointer and Character Strings, Array of Pointers, Pointer as Function Arguments, Functions Returning Pointers, Pointers to Functions, Pointers and Structures 	
Unit-V: File Management in C and Dynamic Memory Allocation	9 hrs

Course Content	Hours
<ul style="list-style-type: none"> File Management in C <ul style="list-style-type: none"> Defining and opening a file, closing a file, input/output operations on files, error handling during I/O operations, random access files, command line argument Dynamic Memory Allocation: Allocating a block of memory: malloc, Allocating Multiple Blocks of Memory : calloc, releasing the used space: Free, Altering the size of a Block: realloc, The Pre-processor : <ul style="list-style-type: none"> Introduction, macro substitution, files inclusion, compiler control directives, ANSI additions 	

Text books:

- C How to Program by Paul Deitel and Harvey Deitel
- Computer Systems: A Programmer's Perspective by Randal E. Bryant and David R.O'Hallaron

Reference books:

- The C Programming Language by Brian Kernighan and Dennis Ritchie
- C Programming: A Modern Approach by K.N. King
- C Primer Plus by Stephen Prata
- Expert C Programming: Deep C Secrets" by Peter van der Linden
- Head First C by David Griffiths and Dawn Griffiths

Pedagogic tools:

- Chalk and Board
- Power point presentation
- Seminar
- Videos
- Lectures and demonstrations
- Flipped classroom approach
- Peer instruction
- Collaborative learning
- Problem-based learning

Methods of Assessment & Tools:

Components of CIA: 30 marks

Sr. No.	Component	Content	Duration (if any)	Marks	Sub Total
A	Test 1	Any 2 Units	1 ^{1/2} hours	5 (Set for 30)	20
	Test 2	Remaining 3 Units	2.5 hours	15 (Set for 45)	
B	Assignment	-	-	5 (20 marks)	10
	Class activity	-	-	5 (20 marks)	
Grand Total					30
Assignment		<ul style="list-style-type: none"> Abstract and executive summary Case study writing Concept mapping Student generated handbook Presentations 			
Class activity		<ul style="list-style-type: none"> Quizzes Poster Presentation Group Discussion 			