

## Programming and Problem Solving Through C

<b>Contents</b>	<b>Lecture No</b>
<b>Overview of C.</b>	
Structure of a C program, Data types, Data types-int, float, char, double, void.	(1)
Data structures, Constants and Variables, Variable declaration: Integer, real, float, character, logical variables, string variables., Constants.	(2)
Operators and Expressions: Arithmetic operators, Relational operators, Logical operators, Expressions. Exchanging value of two variables.	(3)
Control Constructs: If-then, for, while.	(4)
Control Constructs: do-while, switch statements, break and continue, exit() function, go to and label. Summation of set of numbers. Evaluate 'sin x' as sum of series. Test whether a number is prime, Greatest Common Divisor, square root of a number.	(5-6)
Arrays: Array declaration, One and two dimensional arrays.	(7)
Largest number in an array, reverse order of elements of array, multiplication of two matrices print elements of upper triangular matrix. Transpose of a square matrix, Representation of sparse matrix	(8-9)
Functions-Fundamentals: General form, Function arguments, Return value.	(10)
Basic I/O: Formatted Input/Output, Unformatted Input/Output.	(11)
Program Design Examples: Summation of a set of numbers.	(12)
Exercises:- Generation of Fibonacci sequence, Generation of positive Prime numbers, Finding kth smallest element.	(13)
Advanced Features: Type modifiers and storage class specifiers for data types, Bit operators, ? operator, & operator, * operator, Type casting, type conversion.	(14)
Scope Rules: Local and Global variables, Scope rules of Functions.	(15)
Functions: Parameter passing; call-by-value and call-by-reference, Calling functions with Arrays, Argc and Argv. Organize numbers in ascending order, evaluate a polynomial.	(16-17)
Recursion: Basic concept, Design examples, Tower of Hanoi, Recursive Quick sort. Decimal to binary conversion, factorial of a number, reversing digits of an integer.	(18-19)
<b>First Test</b>	(20)
Pointers: the & and * operators*, Pointer expression, Pointer assignments, Pointer arithmetic. Pointer comparison, The dynamic allocation functions-malloc and calloc, Pointer Vs Arrays.	(21-22)
Arrays of Pointers, Pointers to Pointers.	(23)
Initialising Pointers, Pointers to Functions.	(24)

Function returning Pointers, Functions with variable number of Arguments.	(25)
Structures: Basics of Structures, Declaring a Structure, Referencing Structure elements.	(26)
Array of Structures, Passing Structures to Functions, Passing entire Structure to Functions.	(27-28)
Structure Pointers, Declaring a Structure Pointer. Using Structure Pointers, Arrays and Structures within Structures.	(29-30)
Unions: Declaration, Uses, Enumerated data-types, typedef.	(31)
Algorithms, linked list insertion, deletion and search.	(32-33)
File Handling: The file Pointer, file accessing Functions, fopen, fclose, putc, getc, fprintf, fscanf, fseek, fiell etc., different modes for opening files.	(34-35)
C Preprocessor: #define, #include, #undef, #undef, # conditional compilation directives: #if, #else, #else, #elif, #endif, #ifdef, and #ifndef.	(36)
C Standard Library and Header files, Headerfiles, stdio.h, ctype.h, string.h, math.h, stdlib.h, stdarg.h, time.h etc. Standard library functions, String functions, Mathematical functions.	(37-38)
Date and time functions, Variable argument List functions.	(39)
<b>Practical &amp; Final Test</b>	(40)