**1. Introduction to C Programming**

1. History and Features of C
2. Setting up the development environment (IDE, Compiler)
3. Basic structure of a C program
4. Writing your first C program

**2. Data Types and Variables**

1. Primitive Data Types (int, float, char, double)
2. Declaring and initializing variables
3. Constants and Literals
4. Type Conversions and Typecasting

**3. Operators and Expressions**

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Bitwise Operators
5. Assignment Operators
6. Conditional (Ternary) Operator

**4. Input and Output in C**

1. printf() and scanf()
2. gets() and puts() functions
3. Handling different data types in I/O functions

**5. Control Structures**

1. if, else-if, and else statements
2. switch-case statement
3. Looping Structures: for, while, and do-while loops
4. break, continue, and goto statements

**6. Functions**

1. Defining and declaring functions
2. Function arguments and return values
3. Call by value and call by reference
4. Recursion
5. Scope and Lifetime of Variables

**7. Arrays**

1. Introduction to arrays
2. One-dimensional arrays
3. Multi-dimensional arrays
4. Array operations (traversal, insertion, deletion)
5. Passing arrays to functions

**8. Strings**

1. Introduction to strings
2. String functions from **<string.h>**
3. String manipulation and operations
4. Passing strings to functions

**9. Pointers**

1. Introduction to pointers
2. Pointer arithmetic
3. Pointers and arrays
4. Pointers and functions
5. Pointers to pointers

**10. Structures and Unions**

1. Defining and using structures
2. Structure arrays
3. Structure pointers
4. Unions and their use cases
5. Difference between Structures and Unions

**11. Dynamic Memory Allocation**

1. malloc(), calloc(), realloc(), and free()
2. Memory management using pointers
3. Handling memory leaks

**12. File Handling**

1. Introduction to File I/O
2. Opening, reading, writing, and closing files
3. File modes and error handling
4. Random access in files

**13. Preprocessors and Macros**

1. #define and macros
2. #include and file inclusion
3. Conditional Compilation (#ifdef, #ifndef, #endif)
4. Using #pragma and other directives

**14. Advanced Topics**

1. Command Line Arguments
2. Variable Arguments (stdarg.h)
3. Typedef and Enumerations
4. Introduction to Linked Lists
5. Basic Concepts of Data Structures (Stacks, Queues, Trees)

**15. Project Development**

1. Writing and organizing multi-file programs
2. Debugging and testing C programs
3. Developing a small project (e.g., calculator, simple game, student database)