C Programming

Module 1: Introduction to C

- History and importance of C
- Features of C language
- Structure of a C program
- Compilation and execution process
- Setting up a C environment (GCC, IDEs like Code::Blocks or VS Code)
- First program: "Hello, World!"

Module 2: Basics of C Programming

- Keywords and identifiers
- Data types and variables
- Constants and literals
- Operators:
  - Arithmetic
  - Relational
  - Logical
  - Bitwise
  - Assignment
  - Increment/Decrement
  - Conditional (Ternary)
- Type conversion and typecasting

Module 3: Control Structures

- Decision making:
  - `if`, `if-else`, `else-if`, `switch-case`
- Loops:
  - `for`, `while`, `do-while`
- Jump statements:
  - `break`, `continue`, `goto`, `return`

Module 4: Functions

- Function declaration, definition, and calling
- Function arguments and return values
- Scope and lifetime of variables
  - Local, Global, Static
- Recursion
- Inline functions (conceptual)

Module 5: Arrays and Strings

- One-dimensional arrays
- Two-dimensional arrays (matrices)
- Multidimensional arrays
- Introduction to strings
- String handling functions (`strlen`, `strcpy`, `strcmp`, etc.)
- Character arrays vs string literals

Module 6: Pointers

- Introduction to pointers
- Pointer arithmetic
- Pointers and arrays
- Pointers to pointers
- Functions and pointers
- Dynamic memory allocation (`malloc`, `calloc`, `realloc`, `free`)
- Memory management best practices

Module 7: Structures and Unions

- Defining and using structures
- Nested structures
- Arrays of structures
- Pointers to structures
- Structures and functions
- Unions and differences from structures
- `typedef` and enumerations

Module 8: File Handling

- File operations: `fopen`, `fclose`, `fscanf`, `fprintf`, `fread`, `fwrite`
- File modes (`r`, `w`, `a`, etc.)
- Text files vs binary files
- File pointers and error handling
- Command-line arguments

Module 9: Preprocessor and Macros

- Introduction to preprocessor
- `#define`, `#include`, `#undef`, `#ifdef`, `#ifndef`
- Macros with and without arguments
- Conditional compilation

Module 10: Advanced Topics (Optional/Extended)

- Bitwise operations and bit fields
- Storage classes: `auto`, `extern`, `static`, `register`
- Variable argument lists (`stdarg.h`)
- Inline assembly (basic intro)
- Linked lists (single, double, circular)
- Stack and queue using arrays or linked lists
- C Standard Library overview

==========

END

==========