Introduction to C Programming

1 Introduction to C Programming

If youre on Xs coding community, youve seen posts shouting: LEARN C, Here's why:

C is used for low-level system programming, including OS kernels like Linux, device drivers, embedded systems in IoT devices and automotive software, high-performance applications like game engines (e.g., Unreal) and databases, and network protocols. Its direct memory management and efficiency make it essential for speed-critical tasks where higher-level languages can't match performance. Learning it alone is tough, but my X series makes it easy with short daily threads. Tdoay we start with Hello, World!. Don't worry there's loads of things to learn even in a simple "Hello, World!" program.

2 Basic Structure of a C Program

A C program is a collection of instructions that the computer follows. Every program has a starting point called the `main` function, where execution begins. Heres a simple C program:

```
#include <stdio.h>
int main()
{
    printf("Hello, World!\n");
    return 0;
}
```

This program prints Hello, World! to the screen. Lets break it down:

- `#include <stdio.h>`: A preprocessor directive that includes the standard input/output library, giving access to the `printf` function for printing text.
- `int main()`: The main function, the entry point of every C program. The `int` means it returns an integer value.
- `{ and }`: Curly braces define the block of code for the `main` function.
- `printf("Hello, World!
n")`: Prints Hello, World! to the screen. The
`n` adds a new line.
- `return 0`: Signals that the program ended successfully.

3 Preprocessor Directives

Preprocessor directives are instructions processed before the program runs. They start with a `#` symbol and are typically placed at the top of the file. The most common directive is `#include`, which brings in libraries like `stdio.h` for input and output functions.

Important: Every instruction (like `printf`) ends with a semicolon (`;`). Its like a period at the end of a sentence.

4 Rules for Writing a C Program

1. Every C program must have a `main` function as the starting point.
2. Preprocessor directives like `#include` start with `#` and do not end with a semicolon.
3. Instructions inside the `main` function (like `printf`) must end with a semicolon (`;`).
4. Curly braces `{}` group the code inside the `main` function.
5. Save C files with a `.c` extension (e.g., `hello.c`).

5 Example with Explanation

Heres the same program with comments to explain each part:

```
#include <stdio.h>         // Include input/output library
int main()                 // Main function, where program starts
{
    printf("Hello, World!\n"); // Print text to screen
    return 0;              // End program successfully
}
```

This program is the foundation for all C programs. In the next lesson, well learn about storing data using variables and constants.