Conventional programming

Conventional programming, also known as traditional programming, refers to the typical software development process where developers write instructions in programming languages to control the behavior of computers. It follows a procedural approach, where the program executes a sequence of commands or steps in a logical flow to accomplish tasks.

Key characteristics of conventional programming include:

1. **Imperative Approach**: The program specifies how the computer should perform tasks step by step. It uses constructs like loops, conditionals, and variables to control execution.
2. **Explicit Control Flow**: Developers define the flow of control, managing how the program transitions between different states or operations (e.g., through `if` statements, `while` loops, etc.).
3. **Static Typing**: Conventional programming often relies on statically typed languages (e.g., C, Java, C++), where the type of each variable is defined at compile time.
4. **Sequential Execution**: The code typically runs in a sequential manner unless directed otherwise (e.g., with multithreading or asynchronous programming).

Conventional programming contrasts with more modern or specialized paradigms like functional programming, declarative programming, or event-driven programming. While functional programming focuses on immutability and functions as first-class citizens, conventional programming tends to focus on instructions and the flow of control.

In essence, it's about writing code that tells the computer exactly what to do, step by step.