👨‍💻 Introduction to Programming and C Language

🧠 What is Programming?

- **Programming** is the process of writing instructions for a computer to perform specific tasks.
- These instructions are written using a **programming language**.
- A **program** is a set of these instructions that solve a particular problem.

💡 Why Learn Programming?

- To **automate tasks**
- To **build software or apps**
- To **solve real-life problems using logic**
- To enter careers like:
  - Software Development
  - Data Science
  - Cybersecurity
  - Embedded Systems, and more

💬 What is C Programming?

- **C** is a **procedural programming language**, developed in **1972** by **Dennis Ritchie** at Bell Labs.
- Known as the **"Mother of all modern programming languages."**
- Used in:
  - System software (Operating Systems, Compilers)
  - Embedded systems
  - Game development
  - IoT (Internet of Things)

🔄 How Programming Works

1. **Write Code** in a programming language like C
2. **Compile** the code (turn it into machine language)
3. **Execute** (run) the program
4. **Get Output** or result based on inputs and logic

⚙️ Basic Software Setup (for Students)

You'll need:

- **Text Editor** (Notepad++ or built-in IDE)
- **C Compiler** (GCC or Turbo C++)
- **IDE**: Dev C++, Code::Blocks, or VS Code

📌 Structure of a Simple C Program

```
#include <stdio.h>   // Preprocessor directive
int main() {        // Main function
   printf("Hello, World!");   // Output function
   return 0;       // End of program
}
```

Explanation:

- #include <stdio.h> – Includes standard input/output functions
- int main() – Starting point of every C program
- printf() – Used to display output on screen
- return 0; – Ends the program (returns 0 to the OS)

🖍️ Basic Concepts in C

Concept	Description
Variable	Container for storing data (e.g., int x = 5;)
Data Types	Tell the type of data (int, float, char)
Operator	Symbols used to perform operations (+, -, *, /)
Function	A block of code that performs a task
Statement	Each instruction in C ends with a semicolon ;

🧮 Basic Example: Add Two Numbers

```
#include <stdio.h>


int main() {
   int a, b, sum;
   printf("Enter two numbers: ");
   scanf("%d %d", &a, &b);     // Taking input
   sum = a + b;
   printf("Sum = %d", sum);    // Printing result
   return 0;
}
```

📄 Key Features of C Language

- Fast and efficient
- Portable (runs on different systems)
- Simple and powerful
- Foundation of many modern languages (like C++, Java, Python)

🏳️ What You Will Learn in This Course

1. Basics of C Syntax
2. Data types and variables
3. Operators and expressions
4. Control structures (if-else, loops)
5. Functions
6. Arrays and strings
7. Pointers and memory
8. File handling
9. Mini projects (e.g., calculator, student management)

📌 Important Terms

TERM	MEANING
COMPILER	Converts code into machine-readable language
SYNTAX	The grammar/rules of the programming language
DEBUGGING	Finding and fixing errors in the code
EXECUTION	Running the program and getting the output

🧠 Homework / Practice Tasks (First Class)

1. Install C compiler or IDE (like Dev C++ or Code::Blocks)
2. Type and run the Hello, World! program
3. Try changing the message in printf()
4. Write a program that adds two numbers

🎓 Final Thoughts for Day 1

- Programming is like solving puzzles.
- Think logically, break problems into small steps.
- Don’t worry about mistakes—**errors help you learn.**

- Practice is the key to becoming a good programmer.
