📘 SECTION 1: What is Programming? | C Overview, Setup (IDE, Compiler) | Hello World

🟢 BASIC LEVEL

✅ Explanation

- **What is Programming?**
  Programming is the process of writing instructions for a computer to perform specific tasks.
- **What is a Programming Language?**
  A language used to communicate with computers. Example: C, Python, Java.
- **Why learn C?**
  C is powerful, close to hardware, used in systems like operating systems, embedded systems.
- **Tools Setup:**
  - **IDE:** Code::Blocks, Dev C++, or Visual Studio Code
  - **Compiler:** GCC (MinGW for Windows)

💡 Example

```c
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

✍️ Classwork

1. Define “Programming” in your own words.
2. List 3 reasons why C is important.
3. Write and run your first "Hello, World" program in C.

🌍 Real-world Use Case

- The C language is used in the development of operating systems like **Linux** and embedded devices like **TV remotes and microwave controllers**.

🟡 INTERMEDIATE LEVEL

✅ Explanation

- **How Programming Works:**
  - Write code → Compile → Execute → Output
  - Compiler translates human-readable code to machine code.
- **Breakdown of Hello World Code:**
  - `#include <stdio.h>` : Header file for input/output
  - `int main() {}` : Entry point of the program
  - `printf()` : Function to print text
  - `return 0;` : Exit status of the program
- **IDE vs Compiler:**
  - IDE: Development environment (UI + tools)
  - Compiler: Translates code (GCC, Clang)

💡 Example

Explain what happens if:

- You forget a semicolon
- You write `Printf` instead of `printf`
- You miss the `return 0;` line

✍️ Classwork

1. Add a second `printf()` line to say your name.
2. Modify the code to print:

```
Hello, World!
My name is ___.
I love programming!
```

🌍 Real-world Use Case

- In ATMs and traffic light systems, the logic is often programmed in C due to its **speed and efficiency**.

🔴 ADVANCED LEVEL

✅ Explanation

- **Compilation Process:**
  - **Preprocessing → Compilation → Assembly → Linking**
  - Understanding how `gcc` handles code internally
- **Command-line Compilation:**
  - Write code in any editor and compile using:

```
gcc hello.c -o hello
./hello
```

- **Return Codes & Exit Status:**
  - `return 0;` means successful execution
  - `return 1;` or others can indicate different error states

💡 Example

```c
#include <stdio.h>

int main() {
    int code = 0;
    printf("Hello, World!\n");
    return code;
}
```

- Replace `code = 0;` with different values and observe behavior

✍️ Classwork

1. Write a C program that prints:
   - Your name
   - Current year
   - One programming goal
2. Compile using command line (if on Mac/Linux/WSL)

🌍 Real-world Use Case

- Many **firmware systems** used in **medical devices, robots, and drones** are programmed in C for direct memory access and high performance.

✅ Summary Checklist

Topic	Covered
What is Programming?	✅
Why Learn C?	✅
Setup (IDE & Compiler)	✅
Hello World in C	✅

Topic	Covered
Compilation Concepts	✅
Real-world Applications	✅