Day 2: Introduction to C++ Programming

📅 **Date:** 08-05-2025

🛠️ 1. Setting up C++ Environment:

- Installed a C++ compiler (like **g++**) and all necessary packages.
- Configured environment variables correctly to recognize compiler commands in terminal.

🧠 2. Understanding Basic C++ Code:

```
#include <iostream>
```

- # is called the **preprocessor directive** – it runs before compilation.
- `<iostream>` is the **input/output stream** header that gives access to `cout`, `cin`, etc.

🏗️ 3. Structure of a Simple C++ Program:

```
int main() {
    return 0;
}
```

- `main()` is the **starting point** of every C++ program.
- `int` shows the return type – returning `0` means the program ran successfully.
- `return 0;` is a good habit but not strictly required.

🖨️ 4. Printing to the Console:

```
#include <iostream>
int main() {
    std::cout << "Hello World!"; // This is a comment
    return 0;
}
```

- `std::cout` is used to **print output**.
- `<<` is called the **insertion operator**.
- `//` begins a **single-line comment**.
- Every C++ statement ends with a `;` (semicolon).

🧩 5. Recommended VS Code Extensions:

- ✅ **C/C++** – provides syntax support and IntelliSense.
- ✅ **ErrorLens** – highlights compiler errors inline.

🖥️ 6. How to Run Your C++ Program in VS Code:

1. Open terminal (press **Ctrl** + `).
2. Compile the code:
    - This creates an executable:
        - `a.exe` on Windows
        - `a.out` on Mac/Linux
3. Run the program:

```
bash
CopyEdit
./a.exe   # or ./a.out
```

🚀 7. Using Namespace to Simplify Code:

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello World!";
    return 0;
}
```

- `std` is the **standard namespace** in C++.
- `using namespace std;` lets you skip writing `std::` each time.