C++ Lesson: Enums (Enumerations)

1. What is an Enum?

An **enum** (short for *enumeration*) is a **user-defined data type** that consists of a set of **named integer constants**.

Enums make code more readable and manageable by allowing you to use **names** instead of numbers for related constant values.

2. Why Use Enums?

- To represent a **fixed set of related values**
- To improve **code readability**
- To prevent the use of arbitrary numbers (magic numbers)

3. Basic Syntax

```
enum EnumName {
    VALUE1,
    VALUE2,
    VALUE3
};
```

Behind the scenes, the compiler assigns each value an integer:

- `VALUE1` → 0
- `VALUE2` → 1
- `VALUE3` → 2
  ...and so on.

Example:

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

enum Day { Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday };
```

```
int main() {
    Day today = Friday;

    if (today == Friday) {
        cout << "It's almost the weekend!" << endl;
    }

    return 0;
}
```

4. Assigning Custom Values

You can also assign specific integer values:

```
enum Level {
    LOW = 1,
    MEDIUM = 5,
    HIGH = 10
};
```

If you don’t assign a value, C++ will continue from the last assigned value.

Example:

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

enum Priority {
    LOW = 1,
    MEDIUM,     // becomes 2
    HIGH = 10,
    CRITICAL    // becomes 11
};

int main() {
    cout << LOW << endl;       // 1
    cout << MEDIUM << endl;    // 2
    cout << HIGH << endl;      // 10
    cout << CRITICAL << endl;  // 11
    return 0;
}
```

---

5. Enums and Switch Statements

Enums work great with `switch` statements for cleaner code.

✅ Example:

```
enum Status { PENDING, IN_PROGRESS, COMPLETED };

Status taskStatus = IN_PROGRESS;

switch (taskStatus) {
    case PENDING:
        cout << "Task not started." << endl;
        break;
    case IN_PROGRESS:
        cout << "Task is in progress." << endl;
        break;
    case COMPLETED:
        cout << "Task completed!" << endl;
        break;
}
```

❌ 6. Common Mistakes

Mistake	Why it's wrong
Assigning a string to an enum	Enums only hold integers
Expecting enum names to be printed	You must write a function to convert enums to strings manually

---

🛠️ Student Task

✍️ Write a C++ program that:

1. Declares an enum called `TrafficLight` with values `RED`, `YELLOW`, and `GREEN`
2. Assigns one of them to a variable
3. Uses a `switch` statement to print:
    - “Stop” for RED
    - “Slow down” for YELLOW
    - “Go” for GREEN

📌 Bonus: Use the enum in a simple logic flow, like simulating a traffic light system that changes based on user input.