📘 Serializing C++ Classes to a CSV File

📃 What is Serialization?

Serialization is the process of converting an object's state into a format that can be stored (e.g., in a file) or transmitted (e.g., over a network) and later reconstructed. In the context of CSV, serialization means writing the object's data members as a comma-separated line in a `.csv` file.

🧱 Steps to Serialize a Class to CSV

1. **Define the class** with the data members you want to serialize.
2. **Define a member function** like serialize() that writes directly to a stream or file.
3. **Open a file using** `std::ofstream`.

🧱 Serializing a Class

1. Define the Class

```cpp
#include <iostream>
#include <fstream>
#include <string>

class Student {
    std::string name;
    int age;
    double grade;
public:

    Student(std::string n, int a, double g) : name(n), age(a), grade(g) {}

    void SerializeCSV(std::ofstream& outFile,char delimiter) const;
};

void Student::SerializeCSV(std::ofstream& outFile,char delimiter) const
{
    outFile << name << delimiter << age << delimiter << grade;
}
```

2. Call Serialize on the Student objects

```cpp
#include <iostream>
#include <fstream>
#include <string>


int main() {
    // Create student objects
    Student s1("Alice", 20, 88.5);
    Student s2("Bob", 21, 91.0);

    // Open CSV file
    std::ofstream outFile("students.csv");

    if (outFile.is_open()) {

        //call SerializeCSV on each Student object
        s1.SerializeCSV(outFile, ';');
        outFile << '\n';
        s2.SerializeCSV(outFile, ';');

        std::cout << "Data serialized successfully!" << std::endl;
    } else {
        std::cout << "Unable to open file" << std::endl;
    }

    outFile.close();

    return 0;
}
```

📄 Output (`students.csv`)

```
Alice;20;88.5
Bob;21;91.0
```

🧠 Key Concepts

- **File I/O**: `std::ofstream` is used to write to a file.
- **CSV Format**: Fields are separated by a delimiter, and each object is written on a new line.

✅ Best Practices

- Prefer to write a **header row** for clarity.
- Ensure **proper formatting** of fields (e.g., escaping commas if needed).
- Use **exception handling** or file checks to ensure robust file operations.

🎯 Quiz!

Here's a short quiz on the topic: quiz

Footer Separator

🔭 Markdown Viewer

How to view the markdown files in a browser...

- Markdown Viewer

🧠 Lecture Practices

Here are the lecture Practices...

- Day 10
- Day 11

🔍 Lecture Quizzes

Here are the lecture quizzes...

- Day 10
- Day 11

Weekly Topics

Here are the topics for the week...

- CSV
- Writing CSV
- Reading CSV
- Serializing
- Deserializing