Day 101/180 Static data member

1: What is the const keyword in Classes, Create a Customer class and Use Const in it.

In C++, the const keyword indicates a variable, function, or class method should not be modified. It provides a way to specify that certain operations are not allowed, helping to enforce immutability and const-correctness in your code.

For more: https://www.javatpoint.com/const-keyword-in-cpp

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

using namespace std; // Adding the 'using namespace std;' directive

// Define the Customer class
class Customer {
private:
    // Constant member variables
    const string customerId;
    static const int maxAllowedAge = 100; // Static constant member variable

public:
    // Constructor with initialization list for constant member variables
    Customer(const string& id, int age) : customerId(id) {
        // Validate age to ensure it's within allowed range
        if (age < 0 || age > maxAllowedAge) {
            cerr << "Invalid age for customer " << customerId <<
```

```
endl;
            exit(EXIT_FAILURE);
        }
    }

    // Constant member function that does not modify the object's
state
    void displayInfo() const {
        cout << "Customer ID: " << customerId << endl;
    }

    // Function with a constant parameter
    void setAddress(const string& address) {
        // 'address' is treated as read-only within this function
        // ... (implementation to set the address)
    }

    // Static function with a constant local variable
    static void displayMaxAllowedAge() {
        const int localVar = maxAllowedAge;
        cout << "Maximum allowed age: " << localVar << endl;
    }
};

int main() {
    // Create a constant object of the Customer class
    const Customer john("C123", 30);

    // Call the constant member function
    john.displayInfo();

    // Attempting to call a non-const member function on a const
object will result in a compilation error
    // john.setAddress("123 Main St"); // Uncommenting this line will
result in a compilation error
```

```
    // Call a static function using the class name
    Customer::displayMaxAllowedAge();

    return 0;
}
```

2: What is the difference between Encapsulation and Abstraction

Encapsulation: Bundling data and methods into a single unit (class), hiding internal details, and controlling access to the object's state.

Abstraction: Simplifying complex systems by modeling classes based on essential properties and behaviors, ignoring non-essential details, and providing a high-level view of the system's structure.

Read Here:
https://www.geeksforgeeks.org/difference-between-abstraction-and-encapsulation-in-c/