Tutorial 42 - Exercise on C++ Inheritance

Problem Overview

You need to create three classes and perform inheritance as described below:

1. **SimpleCalculator**:
    - Takes input of two numbers using a utility function.
    - Performs basic arithmetic operations: `+`, `-`, `*`, `/`.
    - Displays results using another function.
2. **ScientificCalculator**:
    - Takes input of two numbers using a utility function.
    - Performs four scientific operations of your choice (e.g., power, square root, logarithm, trigonometric functions).
    - Displays results using another function.
3. **HybridCalculator**:
    - Inherits both `SimpleCalculator` and `ScientificCalculator`.
    - Allows access to functionalities of both parent classes.

Questions to Address

1. **Type of Inheritance**: Multiple inheritance (inherits from two base classes).
2. **Mode of Inheritance**: Public mode (for easy access to base class methods).
3. **Object Creation**: Create an object of `HybridCalculator` to display results from both calculators.
4. **Code Reusability**: Achieved through inheritance, reusing methods from `SimpleCalculator` and `ScientificCalculator`.

Implementation:

```cpp
#include <iostream>
#include <cmath>
using namespace std;
class SimpleCalculator {
protected:
    float num1, num2;
public:
    void setNumbers(float a, float b) {
        num1 = a;
        num2 = b;
    }
    void performOperations() {
        cout << "Basic Arithmetic Operations:" << endl;
        cout << "Addition: " << num1 + num2 << endl;
        cout << "Subtraction: " << num1 - num2 << endl;
        cout << "Multiplication: " << num1 * num2 << endl;
        if (num2 != 0) {
            cout << "Division: " << num1 / num2 << endl;
        } else {
            cout << "Division: Undefined (cannot divide by zero)" << endl;
        }
    }
};
class ScientificCalculator {
```

```
protected:
    float num1, num2;
public:
    void setNumbers(float a, float b) {
        num1 = a;
        num2 = b;
    }
    void performScientificOperations() {
        cout << "\nScientific Operations:" << endl;
        cout << "Power (num1^num2): " << pow(num1, num2) << endl;
        cout << "Square Root of num1: " << sqrt(num1) << endl;
        cout << "Logarithm of num1: " << log(num1) << endl;
        cout << "Sine of num1: " << sin(num1) << endl;
    }
};
class HybridCalculator : public SimpleCalculator, public ScientificCalculator {
public:
    void display() {
        performOperations();           // From SimpleCalculator
        performScientificOperations(); // From ScientificCalculator
    }
};
int main() {
    HybridCalculator calc;
    calc.SimpleCalculator::setNumbers(10, 2); // Setting numbers for SimpleCalculator
    calc.ScientificCalculator::setNumbers(10, 2); // Setting numbers for ScientificCalculator
    calc.display();
    return 0;
}

```

Short Notes for Notebook

Steps:

1. **Create Classes**:
    - **SimpleCalculator**:
        - Performs `+`, `-`, `*`, `/`.
    - **ScientificCalculator**:
        - Performs scientific operations: power, square root, logarithm, sine.
    - **HybridCalculator**:
        - Inherits both calculators.
2. **Inheritance**:
    - **Type**: Multiple inheritance.
    - **Mode**: Public.
3. **Reusability**:
    - Common methods are reused in `HybridCalculator`.

Key Code Points:

1. **SimpleCalculator**:
    - Methods: `setNumbers()`, `performOperations()`.
2. **ScientificCalculator**:
    - Methods: `setNumbers()`, `performScientificOperations()`.

3. **HybridCalculator**:
    - Combines methods from both classes using inheritance.

Output Example:

```
Basic Arithmetic Operations:
Addition: 12
Subtraction: 8
Multiplication: 20
Division: 5
Scientific Operations:
Power (num1^num2): 100
Square Root of num1: 3.16228
Logarithm of num1: 2.30259
Sine of num1: -0.54402
```