Tutorial 47 - Solution to Exercise on C++ Inheritance

Problem Statement:

1. Create three classes:
   - **SimpleCalculator**:
     - Input two numbers.
     - Perform basic arithmetic operations: `+`, `-`, `*`, `/`.
   - **ScientificCalculator**:
     - Input two numbers.
     - Perform scientific operations: `cos`, `sin`, `exp`, `tan`.
   - **HybridCalculator**:
     - Inherit both **SimpleCalculator** and **ScientificCalculator**.
     - Display results from both calculators.

Solution

1. SimpleCalculator Class:

```
class SimpleCalculator {
    int a, b;
public:
    void getDataSimple() {
        cout << "Enter the value of a: ";
        cin >> a;
        cout << "Enter the value of b: ";
        cin >> b;
    }
    void performOperationsSimple() {
        cout << "a + b = " << a + b << endl;
        cout << "a - b = " << a - b << endl;
        cout << "a * b = " << a * b << endl;
        cout << "a / b = " << a / b << endl;
    }
};

```

- **Key Points**:
  - Contains private data members: `a`, `b`.
  - Member functions:
    - `getDataSimple()`: Input two numbers.
    - `performOperationsSimple()`: Perform arithmetic operations.

---

2. ScientificCalculator Class:

```
class ScientificCalculator {
    int a, b;
public:
    void getDataScientific() {
        cout << "Enter the value of a: ";
```

```
        cin >> a;
        cout << "Enter the value of b: ";
        cin >> b;
    }
    void performOperationsScientific() {
        cout << "cos(a) = " << cos(a) << endl;
        cout << "sin(a) = " << sin(a) << endl;
        cout << "exp(a) = " << exp(a) << endl;
        cout << "tan(a) = " << tan(a) << endl;
    }
};

```

- **Key Points**:
  - Contains private data members: `a`, `b`.
  - Member functions:
    - `getDataScientific()`: Input two numbers.
    - `performOperationsScientific()`: Perform scientific operations.

---

3. HybridCalculator Class:

```
class HybridCalculator : public SimpleCalculator, public ScientificCalculator {
    // Inherits functionality of both SimpleCalculator and ScientificCalculator
};

```

- **Key Points**:
  - Inherits both **SimpleCalculator** and **ScientificCalculator** using **multiple inheritance**.

---

4. Main Program:

```
int main() {
    HybridCalculator calc;
    cout << "Scientific Calculator Operations:" << endl;
    calc.getDataScientific();
    calc.performOperationsScientific();
    cout << "\nSimple Calculator Operations:" << endl;
    calc.getDataSimple();
    calc.performOperationsSimple();
    return 0;
}

```

- **Key Points**:
  - Creates an object `calc` of type `HybridCalculator`.
  - Calls functions for both calculators using the single object.

Output:

Sample Output 1:

```
Scientific Calculator Operations:
Enter the value of a: 3
```

```
Enter the value of b: 2
cos(a) = -0.989992
sin(a) = 0.14112
exp(a) = 20.0855
tan(a) = -0.142547
Simple Calculator Operations:
Enter the value of a: 10
Enter the value of b: 5
a + b = 15
a - b = 5
a * b = 50
a / b = 2

```

Q&A:

1. **What type of inheritance is used?**
   - **Multiple inheritance**.
2. **What mode of inheritance is used?**
   - **Public** inheritance:
     - `public SimpleCalculator`
     - `public ScientificCalculator`.

Short Notes for Notebook:

1. **Classes**:
   - **SimpleCalculator**:
     - Inputs two numbers.
     - Performs basic arithmetic (`+`, `-`, `*`, `/`).
   - **ScientificCalculator**:
     - Inputs two numbers.
     - Performs scientific operations (`cos`, `sin`, `exp`, `tan`).
   - **HybridCalculator**:
     - Inherits both calculators.
2. **Key Points**:
   - **Multiple Inheritance** is used.
   - Constructors for base classes execute before the derived class.
3. **Main Program**:
   - Object of `HybridCalculator` calls methods from both `SimpleCalculator` and `ScientificCalculator`.
4. **Code**:
   - `class HybridCalculator : public SimpleCalculator, public ScientificCalculator {};`