Tutorial 48 - Constructors in Derived Class (C++)

Key Concepts:

1. **Constructor Execution Order**:
    - **Single Inheritance**: Base class constructor executes before the derived class constructor.
    - **Multiple Inheritance**: Base class constructors execute in the order of inheritance declaration.
    - **Virtual Base Class**: Constructor of the virtual base class executes first, followed by other base class constructors, then the derived class constructor.

Code Examples:

1. Execution Order Cases:

```
// Case 1: Single Inheritance
class B : public A {
    // Order: A() -> B()
};
// Case 2: Multiple Inheritance
class A : public B, public C {
    // Order: B() -> C() -> A()
};
// Case 3: Virtual Base Class
class A : public B, virtual public C {
    // Order: C() -> B() -> A()
};

```

---

2. Parameterized Constructor in Derived Class:

```
class Base1 {
    int data1;
public:
    Base1(int i) {
        data1 = i;
        cout << "Base1 constructor called" << endl;
    }
    void printDataBase1() {
        cout << "Value of data1: " << data1 << endl;
    }
};
class Base2 {
    int data2;
public:
    Base2(int i) {
        data2 = i;
        cout << "Base2 constructor called" << endl;
    }
    void printDataBase2() {
        cout << "Value of data2: " << data2 << endl;
    }
};
```

```cpp
class Derived : public Base2, public Base1 {
    int derived1, derived2;
public:
    Derived(int a, int b, int c, int d) : Base2(b), Base1(a) {
        derived1 = c;
        derived2 = d;
        cout << "Derived class constructor called" << endl;
    }
    void printDataDerived() {
        cout << "Value of derived1: " << derived1 << endl;
        cout << "Value of derived2: " << derived2 << endl;
    }
};

```

Main Program:

```cpp
int main() {
    Derived obj(1, 2, 3, 4); // Create an object of Derived class
    obj.printDataBase1();    // Call Base1 method
    obj.printDataBase2();    // Call Base2 method
    obj.printDataDerived();  // Call Derived method
    return 0;
}

```

Output:

```
Base2 constructor called
Base1 constructor called
Derived class constructor called
Value of data1: 1
Value of data2: 2
Value of derived1: 3
Value of derived2: 4

```

Key Points for Notebook:

1. **Execution Order**:
    - **Single Inheritance**: Base -> Derived.
    - **Multiple Inheritance**: Constructors execute in order of inheritance.
    - **Virtual Base Class**: Virtual base constructor executes first.
2. **Parameterized Constructor in Derived Class**:
    - Use initialization lists to call base class constructors:
      `Derived(int a, int b) : Base1(a), Base2(b) {}`
3. **Important Notes**:
    - Constructors for all base classes are called automatically.
    - Order of inheritance affects constructor execution sequence.

Short Summary:

- **Case 1**: Single inheritance → Base first, then derived.
- **Case 2**: Multiple inheritance → Follow the declared order.
- **Case 3**: Virtual base → Virtual class first, others follow.