**OOPs Advanced concepts comparison**

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Java | Python | C++ |
| Syntax for Defining Inheritance | class Child extends Parent {} | class Child(Parent): | class Child : public Parent {} |
| Multiple Inheritance | Not supported directly. Achieved through interfaces with implements. | Fully supported. Handled using Method Resolution Order (MRO). | Fully supported. Handled via virtual inheritance to avoid the Diamond Problem. |
| Access Modifiers | public, protected, private. No support for private inheritance. | No strict keywords for access modifiers, but uses \_protected and \_\_private name mangling conventions. | public, protected, private inheritance. Control over visibility in the derived class. |
| Method Overriding | Uses @Override annotation. super keyword to call the parent method. | Supports overriding. Uses super() to call the parent method. | Supports overriding with virtual keyword. C++11 onwards allows override keyword for clarity. |
| Constructor and Destructor Calls | Child class constructor calls parent class constructor using super(). No destructors (garbage collector handles cleanup). | super().\_\_init\_\_() or ParentClass.\_\_init\_\_(self) for constructor chaining. Destructors (\_\_del\_\_) rarely needed due to garbage collection. | Parent constructor called automatically, or explicitly in child. Destructors called in reverse order (child first). |
| Abstract Classes and Interfaces | Abstract classes and interfaces enforce method overriding. Interfaces can have default methods from Java 8 onwards. | Abstract classes use the abc module. Methods marked with @abstractmethod. | Abstract classes created using pure virtual functions (= 0). Any class with at least one is abstract. |
| Inheritance Types | Single inheritance only (via extends). Multiple inheritance through interfaces (via implements). | Supports single and multiple inheritance directly. | Supports single and multiple inheritance directly. |
| Diamond Problem | Avoided by design—multiple inheritance not allowed for classes. Interfaces are used to avoid this issue. | Handled using Method Resolution Order (MRO) to ensure the correct method resolution path. | Resolved using virtual inheritance (virtual keyword). |

This table provides a structured overview of how inheritance is managed in Java, Python, and C++.