**👍 Here’s a 15-Day Roadmap to Master Java OOPs (Interview Ready).**  
Each day focuses on one key OOPs concept with **theory + coding practice + mini tasks**.

**🚀 15-Day Java OOPs Roadmap**

**🔹 Day 1 – Encapsulation**

* Learn: Classes, Objects, Access Modifiers (private, public), Getters & Setters.
* Task: Create a User class with name, email, and password (private) → provide getter & setter methods.

**🔹 Day 2 – Inheritance (Single Inheritance)**

* Learn: extends keyword, parent → child relationship.
* Task: Animal → Dog. Parent has eat(), child has bark().

**🔹 Day 3 – Types of Inheritance**

* Learn: Single, Multilevel, Hierarchical, Why Java doesn’t support Multiple Inheritance (via classes).
* Task: Implement CollegeMember → Student → GraduateStudent (multilevel).

**🔹 Day 4 – Method Overriding**

* Learn: Redefining parent’s method in child, @Override annotation.
* Task: Vehicle has speed(), Car overrides with different speed.

**🔹 Day 5 – super Keyword**

* Learn: Calling parent class method/constructor/variable.
* Task: Dog overrides eat(), but also calls parent’s eat() with super.eat().

**🔹 Day 6 – Constructor Chaining**

* Learn: How parent constructor is called before child’s.
* Task: Person → Employee. Print messages from constructors.

**🔹 Day 7 – Polymorphism (Compile-time / Method Overloading)**

* Learn: Method overloading, different signatures.
* Task: Calculator.add(int, int), add(double, double), add(int, int, int).

**🔹 Day 8 – Polymorphism (Runtime / Method Overriding)**

* Learn: Dynamic method dispatch, parent reference holding child object.
* Task: Shape s = new Circle(); → call area().

**🔹 Day 9 – Abstract Classes**

* Learn: abstract keyword, abstract methods.
* Task: Shape as abstract class, subclasses Circle, Rectangle implement area().

**🔹 Day 10 – Interfaces**

* Learn: Interface basics, multiple inheritance using interfaces.
* Task: Create Playable and Recordable interfaces, class MP3Player implements both.

**🔹 Day 11 – Static & Final in OOP**

* Learn: static variables/methods/blocks, final keyword (class, method, variable).
* Task: Demonstrate static counter for number of objects created.

**🔹 Day 12 – Inner Classes**

* Learn: Member inner class, static nested class, anonymous inner class.
* Task: Create an outer class University, with inner class Department.

**🔹 Day 13 – Object Class Methods**

* Learn: toString(), equals(), hashCode(), clone().
* Task: Override toString() in Student class to display details.

**🔹 Day 14 – Java Packages & Access Modifiers**

* Learn: public, protected, private, default across packages.
* Task: Create two packages (school, students) and access classes.

**🔹 Day 15 – Mini Project: OOP Integration**

* Combine all concepts into a small project.
* Example: **Library Management System**
  + Classes: Book, User, Library
  + Features: Borrow/Return books, Track users, Override toString(), Use inheritance for StudentUser and TeacherUser.

⚡ By the end of these 15 days, you’ll be **interview-ready** with Java OOPs, and you’ll have both **theory + practical coding experience**.