OOPS remaining questions JAVA

**Basic-Level OOP Questions (6–10)**

Create a class Car with attributes: make, model, year. Add methods to start, stop the engine, and display info.

Write a class Employee with fields: name, id, salary. Include methods to give a raise and display details.

Create a class Circle with radius as attribute and methods to calculate area and circumference.

Create a class Movie with title, genre, and rating. Add a method to display movie details.

Create a class Laptop with attributes brand, processor, RAM. Add methods to upgrade RAM and display specs.

**Medium-Level OOP Questions (11–20)**

Design a class Account with deposit and withdraw methods, and private balance field (encapsulation).

Create a class Calculator with methods for basic operations: add, subtract, multiply, divide.

Implement a class Library that stores an array of books and has methods to add and display books.

Create a class Temperature that stores temperature in Celsius and converts it to Fahrenheit.

Create a class Animal with a method makeSound(). Inherit it in Dog and Cat classes with specific sounds.

Design a Shape superclass and derive Rectangle, Triangle, and Circle with area methods (inheritance).

Implement a class StudentManager to manage a list of students and find a student by roll number.

Create a class Course with enrolled students, allowing adding/removing students and displaying all.

Create a ShoppingCart class that adds/removes items and calculates total price.

Build a Product class and a Store class to manage a product inventory and search by name.

**Interview-Level OOP Questions (21–30)**

Create an abstract class Vehicle with an abstract method move(). Implement it in Bike and Car.

Implement an interface Playable with method play(). Create classes Football and Cricket implementing it.

Build a BankSystem using interface and classes for SavingsAccount and CurrentAccount.

Create a class Person and override toString(), equals(), and hashCode() methods.

Create a Company class with a list of Employee objects and methods to get highest-paid employee.

Implement method overloading in a MathOperations class with different add() methods.

Demonstrate method overriding with a parent class Bird and a subclass Parrot.

Design a class Logger with a static method to print logs (use of static methods).

Create a Singleton class to ensure only one instance can be created.

Build a StudentGrade class that calculates grade based on marks using conditions and encapsulation.