Java OOP(Object Oriented Programming) Concepts

Last Updated : 25 Jun, 2025

[redacted]

Java Object-Oriented Programming (OOPs) is a fundamental concept in Java that every developer must understand. It allows developers to structure code using **classes and objects**, making it more modular, reusable, and scalable.

The core idea of **OOPs** is to bind data and the functions that operate on it, preventing unauthorized access from other parts of the code. Java strictly follows the DRY (Don't Repeat Yourself) Principle, ensuring that common logic is written once (e.g., in parent classes or utility methods) and reused throughout the application. This makes the code:

- **Easier to maintain:** Changes are made in one place.
- **More organized:** Follows a structured approach.
- **Easier to debug and understand**: Reduces redundancy and improves readability.

In this article, we will explore how **OOPs works in Java** using classes and objects. We will also dive into its **four main pillars of OOPs** that are, **Abstraction, Encapsulation, Inheritance**, and **Polymorphism** with examples.

What is OOPs and Why Do We Use it?

OOPS stands for **Object-Oriented Programming** System. It is a programming approach that organizes code into objects and classes and makes it more structured and easy to manage. A class is a blueprint that defines properties and behaviors, while an object is an instance of a class representing real-world entities.

Example: