**3. Container Concepts in Spring**

**1.Spring IoC (Inversion of Control):**

• **Theory:**

1. **Understanding IoC and how Spring uses it to manage object creation and dependencies:-** IoC:-Inversion of Control (IoC) is a design principle where the control of object creation and dependency management is transferred from the programmer to the Spring framework. Instead of manually creating and managing objects, Spring handles it automatically using an IoC container.

* **How Spring Uses IoC?**

Spring IoC is implemented through **Dependency Injection (DI)**, where dependencies are injected into a class **instead of the class creating them itself**.

1. **Benefits of IoC in application design (loose coupling, modularity, and testability):-** Inversion of Control (IoC) in Spring provides several advantages that improve the design, maintainability, and flexibility of applications.

**1. Loose Coupling**

* IoC removes the direct dependency between classes by injecting dependencies instead of creating them manually.
* Classes depend on **interfaces** rather than specific implementations, making it easy to replace or modify components.

**2. Modularity & Reusability**

* Components can be developed, tested, and maintained separately
* IoC promotes a **modular design**, allowing different parts of the application to evolve independently.
* Easily switch implementations (e.g., replace MySQLDatabase with PostgreSQLDatabase without changing business logic).

**3. Better Testability 🧪**

* Since dependencies are **injected**, we can easily mock them for unit testing.
* No need to initialize entire dependencies manually, reducing test complexity.