SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Coimbatore - 641035.

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++" Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

Department of Computer Applications

Course Code: 23CAT606

Course Name: Java Programming

Unit : V

Topic : Life Cycle of Bean Factory, Explore: Constructor Injection, Dependency Injection

Understanding the Bean Factory Life Cycle

Instantiation

The Bean Factory creates an instance of the bean class.

1

Populating Properties

The Bean Factory sets the properties of the bean instance.

2

BeanNameAware Callback

The Bean Factory calls the setBeanName() method if the bean implements the BeanNameAware interface.

3

Dependency Injection: Constructor Injection

Advantages

Constructor injection ensures that all required dependencies are provided when the bean is created, making the object immutable and thread-safe.

Implementation

Beans are constructed with all their dependencies provided through the constructor, rather than set via setter methods.

Use Cases

Constructor injection is often used for mandatory dependencies and when the bean has a complex lifecycle.

Dependency Injection: Setter Injection

Flexibility

Setter injection allows for more flexibility, as dependencies can be provided or changed at runtime.

Optional Dependencies

Setter injection is useful for optional dependencies that can be injected later in the lifecycle.

Testability

Setter injection makes it easier to test the bean in isolation by mocking dependencies.

Configuration

Setter injection is commonly used in XML-based Spring configuration files.

Dependency Injection: Interface Injection

Decoupling

Interface injection decouples the client from the implementation, promoting flexibility and testability.

Flexibility

Clients can be easily swapped out or updated without modifying the surrounding code.

Composability

Interface injection allows for easy composition of complex objects from smaller, reusable components.

Advantages of Dependency Injection

1 **Loose Coupling**

Dependency injection promotes loose coupling between components, making the system more flexible and maintainable.

2 **Testability**

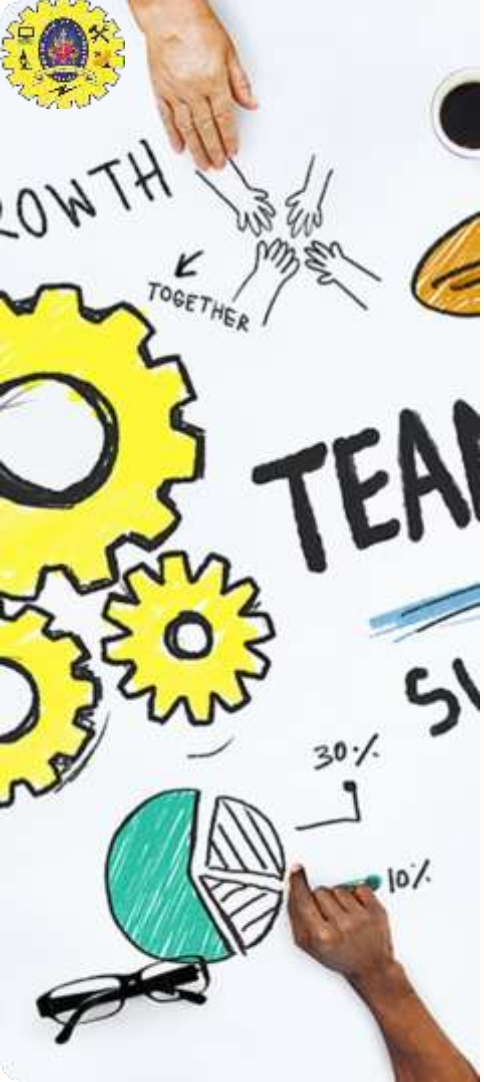
Dependency injection enables easy unit testing by allowing dependencies to be mocked or stubbed.

3 **Configuration Management**

Dependency injection simplifies configuration management, as dependencies can be injected from a central location.

4 **Reusability**

Reusable components can be easily composed together using dependency injection.



Best Practices for Dependency Injectio

1

Identify Dependencies

Clearly identify the dependencies of each component in your application.

2

Choose Injection Type

Decide whether to use constructor, setter, or interface injection based on the specific use case.

3

Use Interfaces

Depend on interfaces rather than concrete implementations to promote flexibility and testability.

Conclusion and Key Takeaways

Dependency Injection	A powerful design pattern that promotes loose coupling and testability.
Bean Factory Life Cycle	Includes instantiation, property population, and various callback methods.
Injection Types	Constructor, setter, and interface injection each have their own advantages.
Best Practices	Identify dependencies, choose the right injection type, and use interfaces.

References

1. "The Complete Reference – Java 2", 8th Edition Tata McGraw Hill, Herbert Schildt
2. w3schools.com
3. "Java Programming form the group up", Tata McGraw Hill, Ralph Bravaco, Shai Simonson