



# 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 : Spring Annotations, Spring AOP

Module





# Spring Annotations and AOP

Explore the powerful annotations and aspects of Spring framework that simplify application development and enable modular, scalable, and testable architecture.



Transactions

Web

Portlet

AOP

Aspects

Instrumentation

Messaging

## @Autowired Annotation

- 1** Automatic Dependency Injection  
The @Autowired annotation injects dependencies automatically, eliminating the need for manual wiring.
- 2** Flexible Injection Points  
It can be used on constructors, setters, and fields to provide flexibility in how dependencies are injected.
- 3** Implicit Naming Conventions  
Spring uses naming conventions to match dependencies, making configuration more intuitive and less verbose.



# Component Annotations

## @Component

Marks a class as a Spring-managed component, allowing it to be automatically detected and registered.

## @Service, @Repository, @Controller

Specialized annotations that indicate the role of the component in the application architecture.

## Stereotypes

These annotations act as stereotypes, providing a consistent way to classify and manage application components.



# Configuration Annotations

## @Configuration

Marks a class as a source of bean definitions to be used by the Spring container.

## @Bean

Indicates a method that instantiates, configures, and initializes a new object to be managed by Spring.

## Programmatic Configuration

These annotations provide a Java-based alternative to the traditional XML-based configuration.



# @Qualifier Annotation

## Disambiguation

The @Qualifier annotation is used to provide additional selection criteria when multiple beans of the same type are available.

## Flexible Matching

Qualifiers enable flexible and dynamic matching of dependencies, enhancing the overall flexibility of the Spring framework.

1

2

3

## Custom Qualifiers

Developers can create their own custom qualifier annotations to further narrow the selection of beans.



# @Value Annotation



## Configuration Injection

The @Value annotation is used to inject external configuration values into Spring-managed beans.



## Dynamic Binding

Values can be bound to fields, constructor arguments, or method parameters, providing flexibility in how they are used.



## Placeholders

Spring supports the use of placeholders in @Value expressions, enabling dynamic and environment-specific configurations.

# AOP

Proxy

Pointcut

## Spring AOP Concepts

1

### Aspects

Modular units that encapsulate cross-cutting concerns, such as logging, security, or transaction management.

2

### Joinpoints

Well-defined points in the application execution flow where an aspect can be applied.

3

### Pointcuts

Expressions that specify which joinpoints an aspect should be applied to.





# Implementing Spring AOP

Declarative

AOP can be implemented declaratively using XML or annotation-based configurations.

Programmatic

AOP can also be implemented programmatically using the Spring AOP API.

Flexibility

Spring AOP provides a flexible and extensible way to add cross-cutting concerns to your application.