**Module 7 Discussion Board – Dependency Injection in Spring**

For this discussion, I decided to look at **Dependency Injection (DI)** in the Spring Framework.

**What:**  
Dependency Injection is a design pattern where the framework takes care of creating and connecting objects instead of developers doing it all manually. This reduces “tight coupling” between classes and makes code easier to update and test. In Spring, DI can be done through constructors, setters, or annotations like @Autowired. The official Spring documentation describes DI as a way to have objects declare their dependencies while the container wires them together at runtime (Spring Framework Documentation, n.d.).

**How:**  
This matters in Java EE development because enterprise applications often have many layers—controllers, services, repositories—that need to work together. Instead of writing boilerplate code to build and pass these dependencies, Spring does the job for us. Personally, I think this is helpful because it feels like plugging in ready-made parts instead of building everything by hand. A recent GeeksforGeeks article points out that DI makes testing easier since you can quickly swap a real dependency for a mock one without changing core code (GeeksforGeeks, 2025).

**Why:**  
The reason DI is so widely used is because it keeps projects maintainable and scalable. It saves time, reduces mistakes, and gives teams the flexibility to reuse code. That’s why DI is considered a best practice in Spring and one of the main reasons the framework became so popular in enterprise Java.

**Example:**

@Component

public class UserService {

private final UserRepository repo;

@Autowired

public UserService(UserRepository repo) {

this.repo = repo;

}

}

Here, Spring automatically injects UserRepository into UserService without needing new UserRepository() in the code.

**References**

GeeksforGeeks. (2025, August 30). *Spring dependency injection with example*. <https://www.geeksforgeeks.org/advance-java/spring-dependency-injection-with-example/>

Spring Framework Documentation. (n.d.). *Dependency injection (DI)*. In *Spring Framework: Core Beans – Factory Collaborators*. <https://docs.spring.io/spring-framework/reference/core/beans/dependencies/factory-collaborators.html>