**WEEK – 3 SPRING CORE MAVEN**

**Implementing Dependency Injection**

**Modify the XML Configuration:**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans   
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="bookRepository" class="org.example.BookRepository"/>  
  
 <bean id="bookService" class="org.example.BookService">  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
</beans>

**Update the BookService Class:**

package com.library;  
  
public class BookService {  
 private BookRepository bookRepository;  
  
 // Setter for Spring DI  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void displayBooks() {  
 System.*out*.println("BookService: Delegating to repository...");  
 bookRepository.printBookList();  
 }  
}

**LibraryManagementApplication.java**

package com.library;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class LibraryManagementApplication {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 BookService bookService = context.getBean("bookService", BookService.class);  
 bookService.displayBooks();  
 }  
}

**Output:**

**BookService: Delegating to repository...**

**BookRepository: Listing all books...**

**Process finished with exit code 0**