**Exercise 11: Implementing Dependency Injection**

**Scenario:**

You are developing a customer management application where the service class depends on a repository class. Use Dependency Injection to manage these dependencies.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **DependencyInjectionExample**.
2. **Define Repository Interface:**
   * Create an interface **CustomerRepository** with methods like **findCustomerById()**.
3. **Implement Concrete Repository:**
   * Create a class **CustomerRepositoryImpl** that implements **CustomerRepository**.
4. **Define Service Class:**
   * Create a class **CustomerService** that depends on **CustomerRepository**.
5. **Implement Dependency Injection:**
   * Use constructor injection to inject **CustomerRepository** into **CustomerService**.
6. **Test the Dependency Injection Implementation:**
   * Create a main class to demonstrate creating a **CustomerService** with **CustomerRepositoryImpl** and using it to find a customer.

**CustomerRepository Interface**: Defines the contract for customer repository operations.

**CustomerRepositoryImpl Class**: Implements the CustomerRepository interface. This class simulates fetching a customer from a database.

**CustomerService Class**: Depends on CustomerRepository to fetch customer details. The dependency is injected through the constructor.

**DependencyInjectionTest Class**: Demonstrates how to use dependency injection by creating instances of CustomerRepositoryImpl and CustomerService, then using CustomerService to fetch and display customer details.