**Exercise 11: Implementing Dependency Injection**

**Step 1: Create a New Java Project**

Create a new Java project named DependencyInjectionExample using your preferred IDE.

**Step 2: Define Repository Interface**

Create an interface CustomerRepository with a method findCustomerById().

// CustomerRepository.java

public interface CustomerRepository {

String findCustomerById(String id);

}

**Step 3: Implement Concrete Repository**

Create a class CustomerRepositoryImpl that implements CustomerRepository.

// CustomerRepositoryImpl.java

public class CustomerRepositoryImpl implements CustomerRepository {

@Override

public String findCustomerById(String id) {

// Simulating database access

return "Customer with ID: " + id;

}

}

**Step 4: Define Service Class**

Create a class CustomerService that depends on CustomerRepository.

// CustomerService.java

public class CustomerService {

private CustomerRepository customerRepository;

// Constructor Injection

public CustomerService(CustomerRepository customerRepository) {

this.customerRepository = customerRepository;}

public String getCustomerDetails(String id) {

return customerRepository.findCustomerById(id);

}

}

**Step 5: Implement Dependency Injection**

Use constructor injection to inject CustomerRepository into CustomerService.

**Step 6: Test the Dependency Injection Implementation**

Create a main class to demonstrate creating a CustomerService with CustomerRepositoryImpl and using it to find a customer.

// DependencyInjectionExample.java

public class DependencyInjectionExample {

public static void main(String[] args) {

// Create a CustomerRepository implementation

CustomerRepository customerRepository = new CustomerRepositoryImpl();

// Inject the repository into the service

CustomerService customerService = new CustomerService(customerRepository);

// Use the service to find a customer

String customerDetails = customerService.getCustomerDetails("12345");

System.out.println(customerDetails);

}

}