**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.

**1. Customer.java**

public class Customer {

private String id;

private String name;

public Customer(String id, String name) {

this.id = id;

this.name = name;

}

public String getId() { return id; }

public String getName() { return name; }

}

**2. CustomerRepository.java – Interface**

public interface CustomerRepository {

Customer findCustomerById(String id);

}

**3. CustomerRepositoryImpl.java – Concrete Implementation**

import java.util.HashMap;

import java.util.Map;

public class CustomerRepositoryImpl implements CustomerRepository {

private Map<String, Customer> customerData = new HashMap<>();

public CustomerRepositoryImpl() {

// Dummy data

customerData.put("101", new Customer("101", "Alice"));

customerData.put("102", new Customer("102", "Bob"));

}

@Override

public Customer findCustomerById(String id) {

return customerData.get(id);

}

}

**4. CustomerService.java – Class That Uses Dependency**

public class CustomerService {

private CustomerRepository repository;

// Constructor Injection

public CustomerService(CustomerRepository repository) {

this.repository = repository;

}

public void displayCustomer(String id) {

Customer customer = repository.findCustomerById(id);

if (customer != null) {

System.out.println("Customer Found: " + customer.getName() + " (ID: " + customer.getId() + ")");

} else {

System.out.println("Customer not found with ID: " + id);

}

}

}

**5. Test.java – Testing Dependency Injection**

public class Test {

public static void main(String[] args) {

// Create the repository

CustomerRepository repository = new CustomerRepositoryImpl();

// Inject the dependency into the service

CustomerService service = new CustomerService(repository);

// Use the service

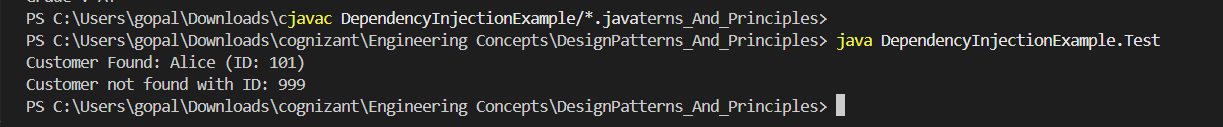
service.displayCustomer("101");

service.displayCustomer("999"); // Not present

}

}

**Output:**

****