**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.

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

Create a new Java project named DependencyInjectionExample.

**Step 2:** Define Repository Interface

//CustomerRepository

interface CustomerRepository{

void findCustomerById(String id);

}

**Step 3:** Implement Concrete Repository

// CustomerRepositoryImpl

class CustomerRepositoryImpl implements CustomerRepository{

private String customerName;

private String customerId;

private String customerDept;

public CustomerRepositoryImpl(String customerName,String customerId,String customerDept)

{

this.customerName = customerName;

this.customerId = customerId;

this.customerDept = customerDept;

}

@Override

public void findCustomerById(String id)

{

if(customerId.equals(id))

{

System.out.println("The customer Id "+id+" belongs to "+customerName);

}

}

}

**Step 4:** Service Class

// CustomerService

class CustomerService implements CustomerRepository{

private List <CustomerRepository> customers = new ArrayList<>();

public void addCustomers(CustomerRepository custrep)

{

customers.add(custrep);

}

@Override

public void findCustomerById(String id){

for(CustomerRepository customer:customers)

{

customer.findCustomerById(id);

}

}

}

**Step 5:** Dependency Injection Implementation

public class DependencyInjection {

public static void main(String[] args) {

CustomerRepository cust1 = new CustomerRepositoryImpl("ABC","101","CSE");

CustomerRepository cust2 = new CustomerRepositoryImpl("DEF","102","ECE");

CustomerRepository cust3 = new CustomerRepositoryImpl("GHI","103","EEE");

CustomerService custservice = new CustomerService();

custservice.addCustomers(cust1);

custservice.addCustomers(cust2);

custservice.addCustomers(cust3);

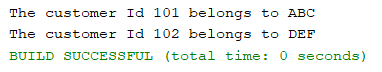
custservice.findCustomerById("101");

custservice.findCustomerById("102");

}

}

**SAMPLE OUTPUT:**

****