**Design Pattern And Principles**

**EXERCISE 11: Implementing the Dependency Injection**

**Source Code**

// Main.java

// Step 2: Repository Interface

interface CustomerRepository {

String findCustomerById(String id);

}

// Step 3: Concrete Repository Implementation

class CustomerRepositoryImpl implements CustomerRepository {

public String findCustomerById(String id) {

// For simplicity, return a fake customer record

return "Customer[ID=" + id + ", Name=Ramin, Email=ramin@example.com]";

}

}

// Step 4: Service Class

class CustomerService {

private final CustomerRepository customerRepository;

// Step 5: Constructor Injection

public CustomerService(CustomerRepository customerRepository) {

this.customerRepository = customerRepository;

}

public void displayCustomer(String id) {

String customer = customerRepository.findCustomerById(id);

System.out.println("Retrieved: " + customer);

}

}

// Step 6: Test the Dependency Injection

public class Main {

public static void main(String[] args) {

// Injecting CustomerRepositoryImpl into CustomerService

CustomerRepository repository = new CustomerRepositoryImpl();

CustomerService service = new CustomerService(repository);

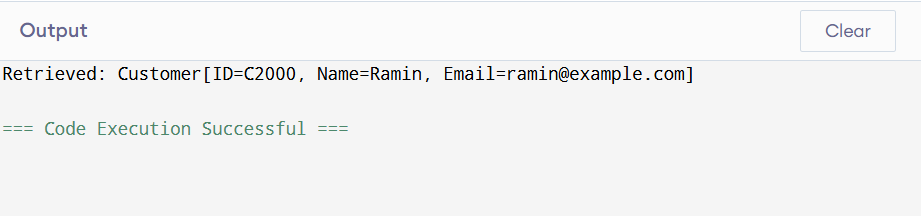
// Use the service to find and display a customer

service.displayCustomer("C2000");

}

}

**OUTPUT**

****