Exercise 11: Customer Search Using Dependency Injection

Approach and Understanding:

In this exercise, I built a **Customer Search Application** where we can type a **Customer ID**

and find if the customer exists.

**Code**

import java.util.Scanner;

class Customer { String id; String name;

Customer(String id, String name) { this.id = id;

this.name = name;

}

public String toString() {

return "ID: " + id + ", Name: " + name;

}

}

interface CustomerRepo { Customer find(String id);

}

class CustomerRepoImpl implements CustomerRepo { Customer[] list = {

new Customer("C1", "Meena"),

new Customer("C2", "Ravi"), new Customer("C3", "Sara")

};

public Customer find(String id) { for (Customer c : list) {

if (c.id.equalsIgnoreCase(id)) { return c;

}

}

return null;

}

}

class CustomerService { CustomerRepo repo;

CustomerService(CustomerRepo repo) { this.repo = repo;

}

void findCustomer(String id) {

Customer c = repo.find(id); if (c != null) {

System.out.println("Found: " + c);

} else {

System.out.println("Not found.");

}

}

}

public class CustomerApp {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

CustomerRepo repo = new CustomerRepoImpl(); CustomerService service = new CustomerService(repo);

System.out.println("Hi! Search customer by ID. Type 'stop' to exit.");

while (true) { System.out.print("\nID to find: "); String id = sc.nextLine().trim();

if (id.equalsIgnoreCase("stop")) { break;

}

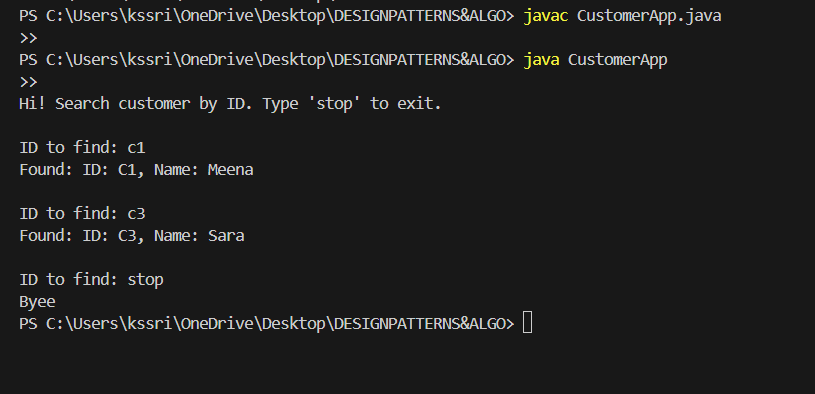
service.findCustomer(id);

}

sc.close(); System.out.println("Byee");

}

}



**OUTPUT:**