

COMP3211 Assignment 2

Name: JIANG GUANLIN

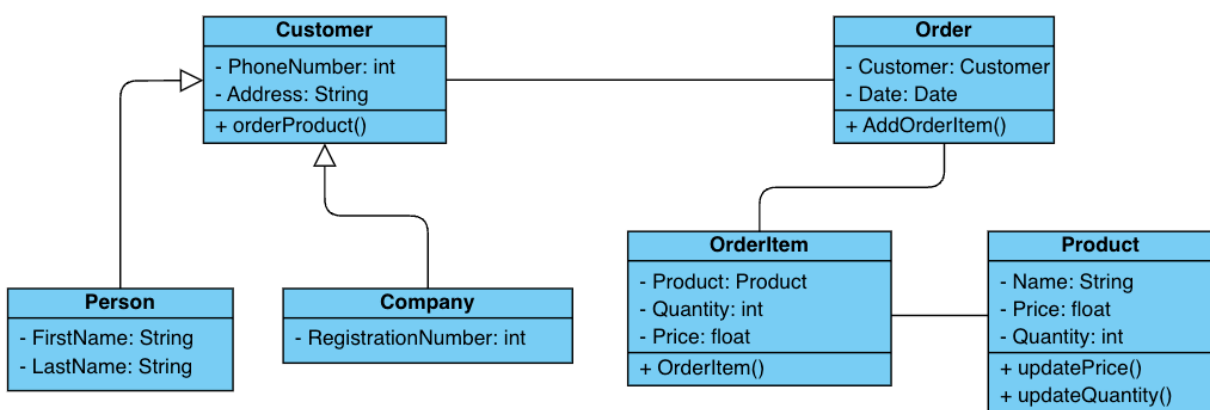
Student ID: 21093962D

1 Class Diagram

- Class: Product
 - Attributes
 - Name: String
 - Price: float
 - Quantity: int
 - Methods
 - updatePrice()
 - updateQuantity()
- Class: Customer
 - Attributes
 - PhoneNumber: int
 - Address: String
 - Methods
 - orderProduct()
- Class: Person (inherits Class Customer)
 - Attributes
 - FirstName: String
 - LastName: String
- Class: Company (inherits Class Customer)

- Attributes
 - RegistrationNumber: int
- Class: Order
 - Attributes
 - Customer: Customer
 - Date: Date
 - Methods
 - AddOrderItem()
- Class: OrderItem
 - Attributes
 - Product: Product
 - Quantity: int
 - Price: float
 - Methods
 - OrderItem()

UML Diagram



2 Layered Architecture

- User Interface Layer
 - (h) Browser UI
 - (i) Mobile UI
- User Communications Authentication & Authorization Layer
 - (a) Mobile Device Management
 - (g) Forms Management
- Information retrieval & Modification Layer
 - (b) Database Search
 - (e) Database Browser
 - (f) Database Query Management
- Transaction management Database Layer
 - (c) Buildings Database
 - (d) Equipment Database
 - (j) Vehicle Database

3 Software Testing

TEST	X	Y	Z	EXPECTEDRESULT	BRANCHES COVERED
Test 1	1	1	-1	false	c2 == true
Test 2	1	1	1	true	c2 == false, c5 == false, c7 == false, c9 == true
Test 3	2	3	4	true	c2 == false, c5 == false, c7 == false, c9 == true
Test 4	0	1	2	false	c2 == true
Test 5	1	2	3	false	c2 == false, c5 == true
Test 6	1	1	3	false	c2 == false, c5 == true
Test 7	1	3	1	false	c2 == false, c5 == false, c7 == true
Test 8	3	1	1	false	c2 == false, c5 == false, c7 == false, c9 == true
Test 9	0	0	0	false	c2 == true

4 Software Maintenance

- Corrective
 - To solve the bugs and errors in the software after deploy, found the problem that affect the software using, and fix it, recover it to the expected functions.
- Adaptive
 - Fix the software to achieve that can use in different environment. Also update the software and the hardware if needed, to make sure the software is suitable for this case.
- Perfective
 - To improve the software performance, efficiency and maintainability, which can continuous optimization, and improve the functions.
- Preventive
 - To solve the problem before the problem pop up when user use it. The code review, refactoring and performance monitoring are more important to checking the software.