

NOTES: Complete Working Code (Zomato)

Importing Libraries

Java

```
import java.time.LocalDateTime;
import java.util.ArrayList;
import java.util.HashMap;
```



Class Zomato:

Java

```
class Zomato {
    public static void main(String[] args) {

    }
}
```

Enum OrderStatus

Java

```
enum OrderStatus {
    ACTIVE, CLOSED, CANCELED,
}
```

Enum PaymentStatus

Java

Articles Read

0 of 1 Complete. (0%)

UNPAID, PENDING, COMPLETED, FILLED, DECLINED, CANCELLED



Articles

Java



Videos

```
enum PaymentMethod {  
    CASH, UPI, CARD, NET_BANKING  
}
```

**Class Person:**

Java

```
class Person {  
    public String name, email, phone;  
    public Address address;  
  
    public Person(String name, String email, String phone, Address address)  
    {  
        this.name = name;  
        this.email = email;  
        this.phone = phone;  
        this.address = address;  
    }  
}
```

Class Address:

Java

```
class Address {  
    public String street, city, state, zipcode, country;
```



Menu

Articles Read

0 of 1 Complete. (0%)

`this.state = state;`

Dash



Articles

}



Videos

Class Customer:

Java



```
class Customer extends Person {  
  
    public ArrayList<Order> orders;  
  
    Customer(String name, String email, String phone, Address address) {  
        super(name, email, phone, address);  
    }  
  
    public void addOrders(Order order) {  
        this.orders.add(order);  
    }  
  
    public Boolean receiveOrder(Order order) {  
        if (!orders.contains(order))  
            return false;  
  
        this.orders.remove((order));  
        return true;  
    }  
}
```

Class DeliveryGuy:

Java



Menu

Articles Read

0 of 1 Complete. (0%)

Dash



Articles

```
public void addDeliveries(Order order) {  
    this.deliveries.add(order);  
}  
  
public Boolean deliverItem(Order order) {  
    if (!deliveries.contains(order))  
        return false;  
    this.deliveries.remove(order);  
    return true;  
}  
}
```



Videos



Class Restaurant:

Java

```
class Restaurant {  
    public String name;  
    public ArrayList<Branch> branches;  
  
    Restaurant(String name, Branch initialBranch) {  
        this.name = name;  
        this.branches = new ArrayList<Branch>();  
        this.branches.add(initialBranch);  
    }  
  
    public void addBranch(Branch branch) {  
        this.branches.add(branch);  
    }  
}
```



Menu

Articles Read

0 of 1 Complete. (0%)

Dash



Articles



Videos

```
public String name;

public Branch(Address address, String name) {
    this.address = address;
    this.name = name;
}

}
```



Class Menu:

Java

```
class Menu {
    public String menuId;
    public ArrayList<FoodItem> menu = new ArrayList<FoodItem>();

    Menu(String id) {
        this.menuId = id;
    }

    public void addFood(FoodItem dish) {
        this.menu.add(dish);
    }
}
```

Class FoodItem:

Java

```
class FoodItem {
```



Menu

Articles Read

0 of 1 Complete. (0%)

Dash



Articles



Videos

```
this.description = description;
this.cost = cost;
}

public void updatePrice(int cost) {
    this.cost = cost;
}

}
```

**Class Order:**

Java

```
class Order {

    public String orderId;
    public Customer customer;
    public DeliveryGuy delivery;
    public LocalDateTime creationOrder;
    public PaymentMethod billingMode;
    public ArrayList<FoodItem> meal;

    public Order(String orderId, Customer customer, DeliveryGuy delivery, Lo
        PaymentMethod billingMode) {
        this.orderId = orderId;
        this.customer = customer;
        this.delivery = delivery;
        this.creationOrder = creationOrder;
        this.billingMode = billingMode;
        meal = new ArrayList<FoodItem>();
    }
}
```



Menu

Articles Read

0 of 1 Complete. (0%)

}

Dash



Articles

Courses

Tutorials

Jobs

Practice

Contests



```
class System {  
    public HashMap<Customer, ArrayList<Order>> customerDB;  
    public HashMap<DeliveryGuy, ArrayList<Order>> deliveryDB;  
  
    public void addCustomerDetails(Customer user, Order order) {  
        customerDB.get(user).add(order);  
    }  
  
    public ArrayList<Order> getCustomerDetails(Customer customer) {  
        return customerDB.get(customer);  
    }  
  
    public void addDeliveryDetails(DeliveryGuy delivery, Order order) {  
        deliveryDB.get(delivery).add(order);  
    }  
  
    public ArrayList<Order> getDeliveryDetails(DeliveryGuy delivery) {  
        return deliveryDB.get(delivery);  
    }  
}
```

Mark as Read

Menu

Report An Issue



If you are facing any issue on this page. Please let us know.

Articles Read

0 of 1 Complete. (0%)

Dash



Articles



Videos



Menu



Articles Read

0 of 1 Complete. (0%)