

Nama : HASBI HASBULLAH

NIM : 312110094

Kelas : TI.21.C.1

☞ Cash.java

```
public class Cash extends Payment {
    private float cashTendered;

    public float getCashTendered() {
        return cashTendered;
    }

    public void setCashTendered(float cashTendered) {
        this.cashTendered = cashTendered;
    }
}
```

☞ Check.java

```
public class Check extends Payment {
    private String name;
    private String bankID;

    public void authorized() {
        System.out.println("Authorizing...");
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getBankID() {
        return bankID;
    }

    public void setBankID(String bankID) {
        this.bankID = bankID;
    }
}
```

☞ Credit.java

```
import java.time.LocalDate;

public class Credit extends Payment {
    private String number;
    private String type;
    private LocalDate expDate;

    public void authorized() {
        System.out.println("Authorizing...");
    }

    public String getNumber() {
        return number;
    }

    public void setNumber(String number) {
        this.number = number;
    }

    public String getType() {
        return type;
    }

    public void setType(String type) {
        this.type = type;
    }

    public LocalDate getExpDate() {
        return expDate;
    }

    public void setExpDate(LocalDate expDate) {
        this.expDate = expDate;
    }
}
```

Customer.java

```
import java.util.ArrayList;
import java.util.List;

public class Customer {
    private String name;
    private String address;
    private List<Order> orderList = new ArrayList<Order>();

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public void addOrder(Order order) {
        this.orderList.add(order);
    }
}
```

Item.java

```
import java.util.ArrayList;
import java.util.List;

public class Item {
    private double shippingWeight;
    private String description;
    private List<OrderDetail> orderDetailList = new ArrayList<OrderDetail>();

    public double getShippingWeight() {
        return shippingWeight;
    }

    public void setShippingWeight(double shippingWeight) {
        this.shippingWeight = shippingWeight;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public void addOrderDetail(OrderDetail orderDetail) {
        this.orderDetailList.add(orderDetail);
    }

    public void getPriceForQuantity() {
        System.out.println("Getting price for quantity...");
    }

    public void getTax() {
        System.out.println("Getting Tax...");
    }

    public void inStock() {
        System.out.println("In Stock!");
    }
}
```

```

import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;

public class Order {
    private LocalDate date;
    private String status;
    private Customer customer;
    private List<Cash> cashList;
    private List<Check> checkList;
    private List<Credit> creditList;
    private List<OrderDetail> orderDetailList;

    public Order() {
        this.cashList = new ArrayList<Cash>();
        this.checkList = new ArrayList<Check>();
        this.creditList = new ArrayList<Credit>();
        this.orderDetailList = new ArrayList<OrderDetail>();
        this.customer = new Customer();
    }

    public LocalDate getDate() {
        return date;
    }

    public void setDate(LocalDate date) {
        this.date = date;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public void addCashList(Cash cash) {
        this.cashList.add(cash);
    }

    public void addCheckList(Check check) {
        this.checkList.add(check);
    }

    public void addCreditList(Credit credit) {
        this.creditList.add(credit);
    }

    public void addOrderDetail(OrderDetail orderDetail) {
        this.orderDetailList.add(orderDetail);
    }

    public void calcSubTotal() {
        System.out.println("Calculating Sub Total...");
    }

    public void calcTax() {
        System.out.println("Calculating Tax...");
    }

    public void calcTotal() {
        System.out.println("Calculating Total...");
    }

    public void calcTotalWeight() {
        System.out.println("Calculating Total Weight...");
    }
}

```

OrderDetail.java

```
public class OrderDetail {
    private int quantity;
    private String taxStatus;
    private Order order;
    private Item item;

    public OrderDetail(Order order) {
        this.order = order;
        this.item = new Item();
    }

    public int getQuantity() {
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    public String getTaxStatus() {
        return taxStatus;
    }

    public void setTaxStatus(String taxStatus) {
        this.taxStatus = taxStatus;
    }

    public void calcSubTotal() {
        System.out.println("Calculating Sub Total...");
    }

    public void calcWeight() {
        System.out.println("Calculating Weight...");
    }

    public void calcTax() {
        System.out.println("Calculating Tax...");
    }
}
```

Payment.java

```
public abstract class Payment {
    private float amount;
    private Order order;

    public Payment() {
        order = new Order();
    }
}
```

```

import java.time.LocalDate;

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

        Customer hasbi = new Customer();
        hasbi.setName("Hasbi hasbullah");
        hasbi.setAddress("Kp. Bulak Kunyit RT.001/RW.002 Desa Muktiwari");
        System.out.println("Pelanggan atas nama " + rio.getName() + " dipersilahkan masuk!");
        System.out.println("Pesanan atas nama " + rio.getName() + " siap dikirimkan ke alamat " + rio.ge

        System.out.println();

        Order gofood = new Order();
        rio.addOrder(gofood);
        gofood.calcSubTotal();
        gofood.calcTax();
        gofood.calcTotal();
        gofood.calcTotalWeight();
        gofood.setDate(LocalDate.now());
        gofood.setStatus("Shipping...");
        System.out.println("Pesanan pada tanggal : " + gofood.getDate());
        System.out.println("Status : " + gofood.getStatus());

        System.out.println();

        OrderDetail detailOrderGoFood = new OrderDetail(gofood);
        detailOrderGoFood.calcSubTotal();
        detailOrderGoFood.calcWeight();
        detailOrderGoFood.calcTax();
        detailOrderGoFood.setQuantity(5);
        detailOrderGoFood.setTaxStatus("Dipotong pajak");
        gofood.addOrderDetail(detailOrderGoFood);
        System.out.println("Jumlah item yang dibeli: " + detailOrderGoFood.getQuantity());
        System.out.println("Status Pajak: " + detailOrderGoFood.getTaxStatus());

        System.out.println();

        Item item = new Item();
        item.getPriceForQuantity();
        item.getTax();
        item.inStock();
        item.setShippingWeight(5.0);
        item.setDescription("Pembelian barang dengan kondisi baru.");
        item.addOrderDetail(detailOrderGoFood);
        System.out.println("Total berat pengiriman sebesar: " + item.getShippingWeight() + "Kg.");
        System.out.println("Deskripsi barang: " + item.getDescription());

        Cash cash = new Cash();
        cash.setCashTendered(10000.0F);
        gofood.addCashList(cash);
        System.out.println("Pembayaran ongkos kirim sebesar: Rp" + (int) cash.getCashTendered());

        System.out.println();

        Check check = new Check();
        check.setName("Hasbi hasbullah");
        check.setBankID("12341234");
        check.authorized();
        gofood.addCheckList(check);
        System.out.println("Pembayaran atas nama " + check.getName() + " dengan IDBank " + check.getBank);

        System.out.println();

        Credit credit = new Credit();
        credit.authorized();
        credit.setNumber("11100040050330");
        credit.setType("Debit");
        credit.setExpDate(LocalDate.now());
        gofood.addCreditList(credit);
        System.out.println("Nomor kartu credit " + credit.getNumber() + " dengan tipe " + credit.getType);

        System.out.println("Terima kasih telah melakukan pesanan, Happy Shopping :)");
    }
}

```