اونيبرسيتي تيكنولوجي مارا

UNIVERSITI TEKNOLOGI MARA

CSC186 – OBJECT ORIENTED PROGRAMMING

LAB ASSIGNMENT 3

NAME : MUHAMMAD REDZA BIN MAHAYADIN

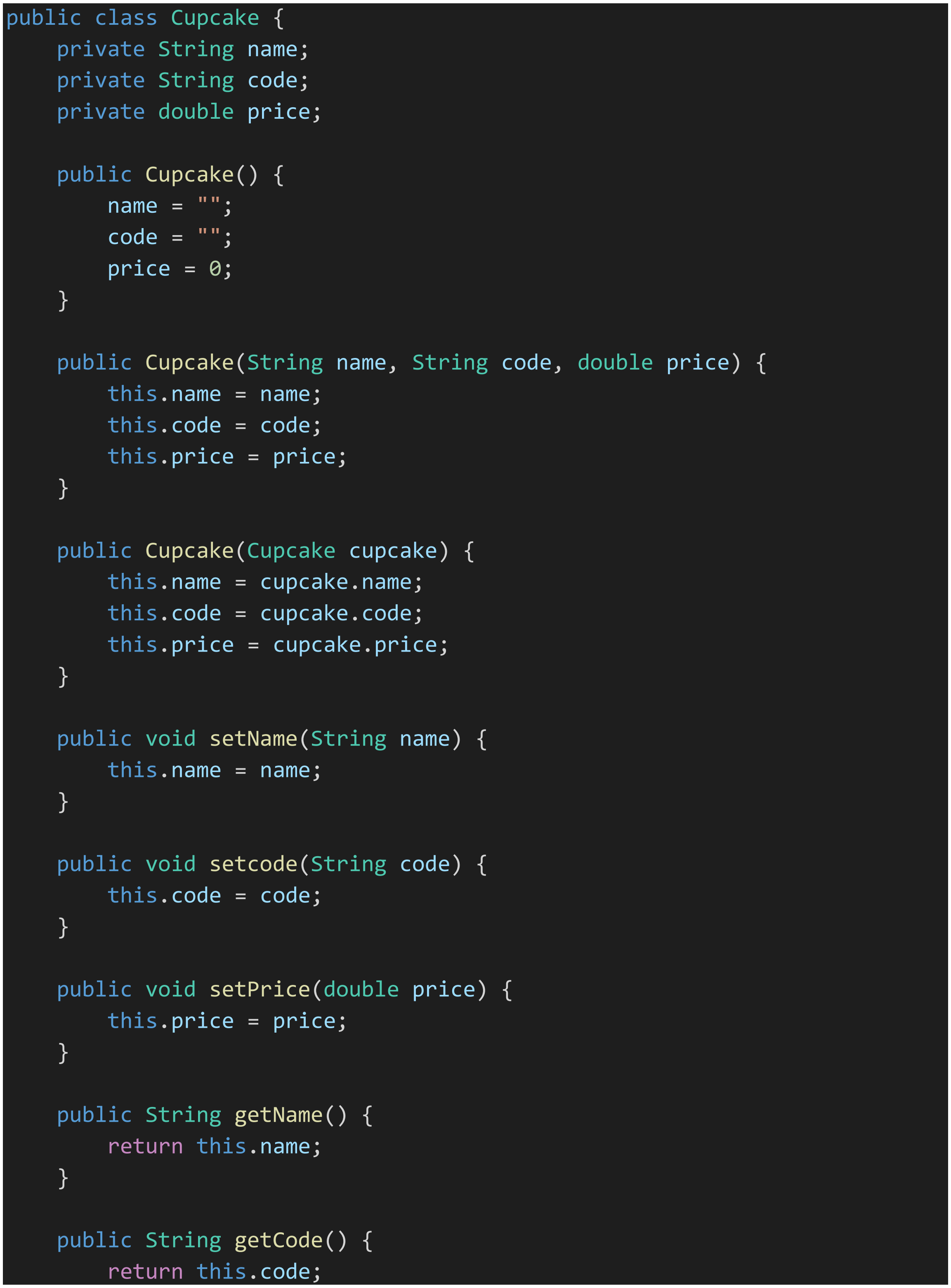
STUDENT ID : 2022676696

GROUP : RCDCS1102B

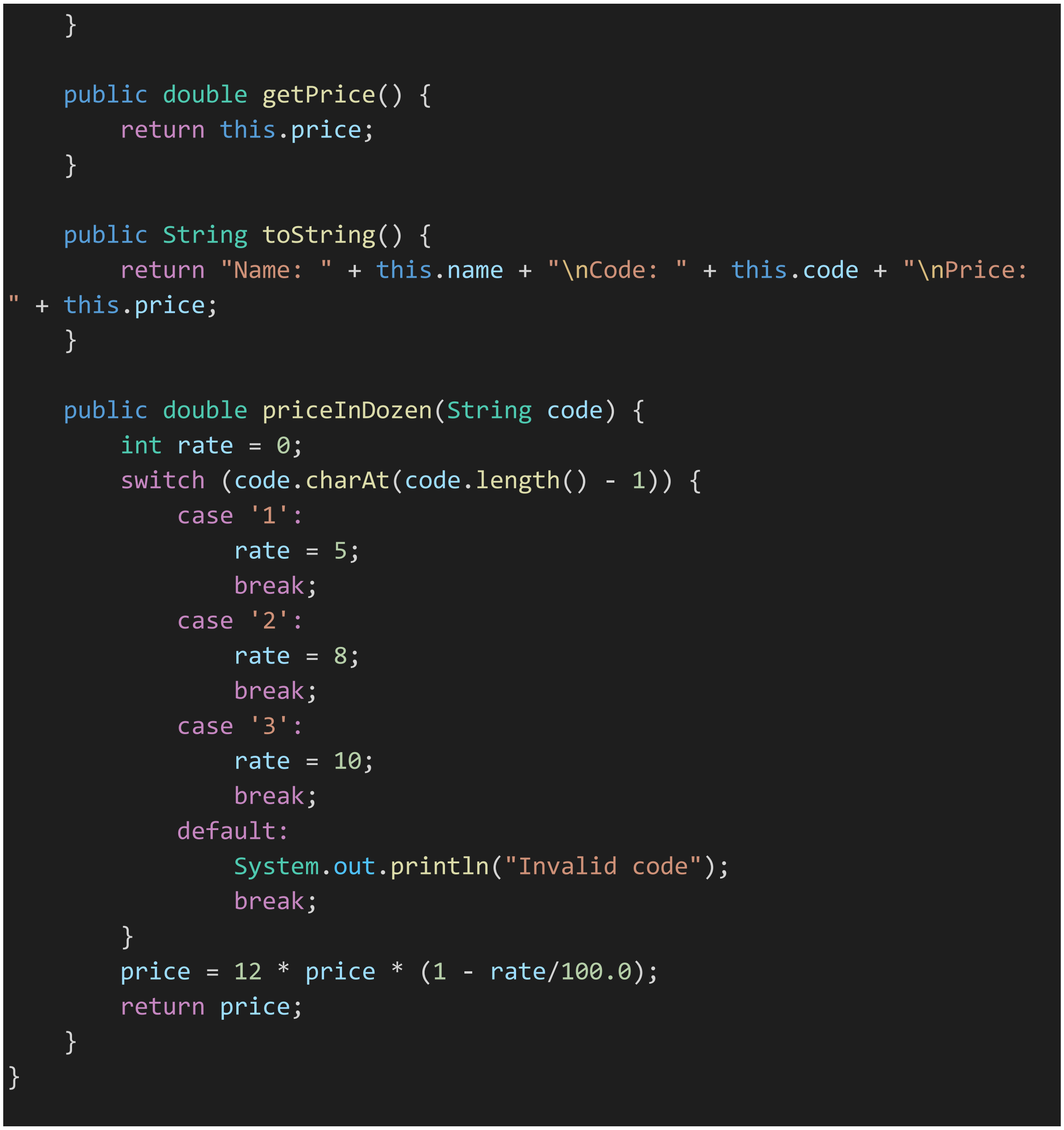
LECTURER : SIR MOHD NIZAM BIN OSMAN

QUESTION 3.1

SOURCE CODE 1.1 : CUPCAKE CLASS

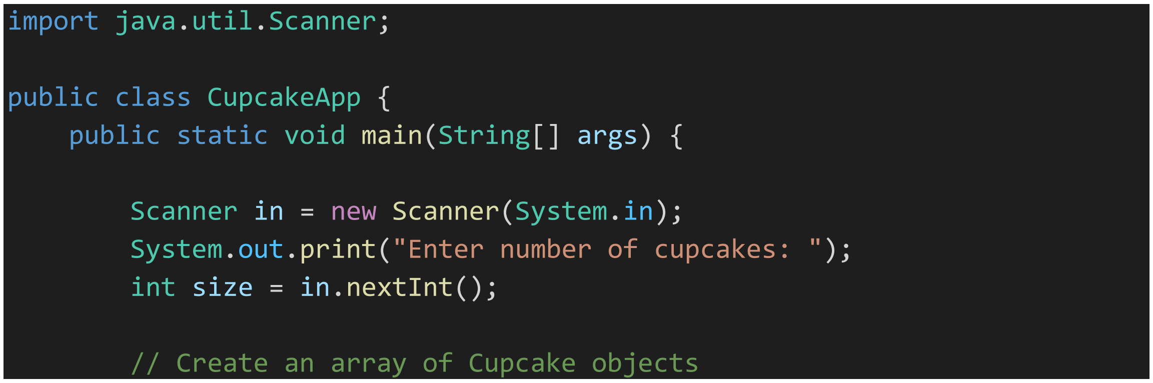


```java
public class Cupcake {
    private String name;
    private String code;
    private double price;

    public Cupcake() {
        name = "";
        code = "";
        price = 0;
    }

    public Cupcake(String name, String code, double price) {
        this.name = name;
        this.code = code;
        this.price = price;
    }

    public Cupcake(Cupcake cupcake) {
        this.name = cupcake.name;
        this.code = cupcake.code;
        this.price = cupcake.price;
    }

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

    public void setcode(String code) {
        this.code = code;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public String getName() {
        return this.name;
    }

    public String getCode() {
        return this.code;
```

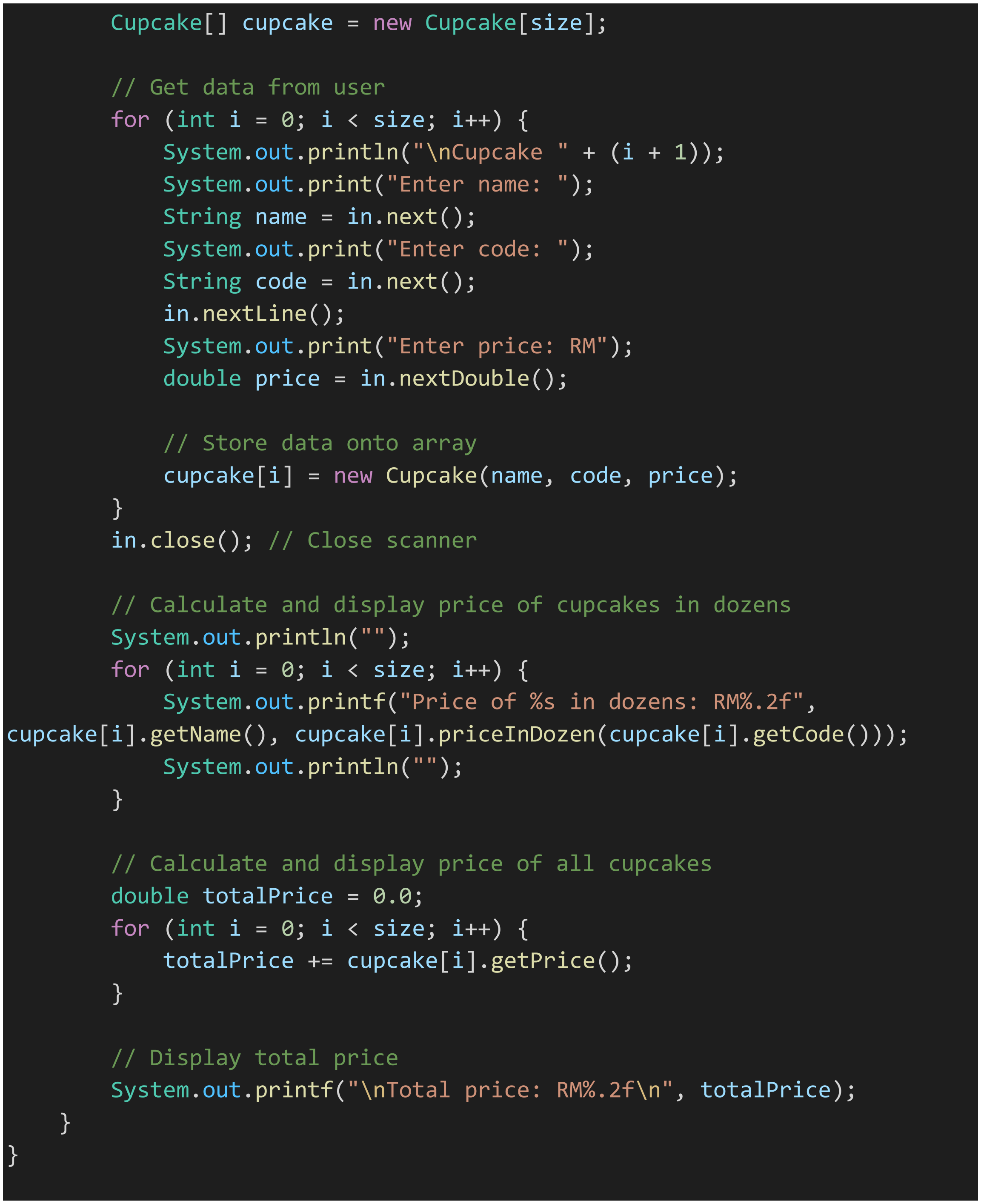


```
    }

    public double getPrice() {
        return this.price;
    }

    public String toString() {
        return "Name: " + this.name + "\nCode: " + this.code + "\nPrice: 
" + this.price;
    }

    public double priceInDozen(String code) {
        int rate = 0;
        switch (code.charAt(code.length() - 1)) {
            case '1':
                rate = 5;
                break;
            case '2':
                rate = 8;
                break;
            case '3':
                rate = 10;
                break;
            default:
                System.out.println("Invalid code");
                break;
        }
        price = 12 * price * (1 - rate/100.0);
        return price;
    }
}
```

SOURCE CODE 1.2 : CUPCAKEAPP

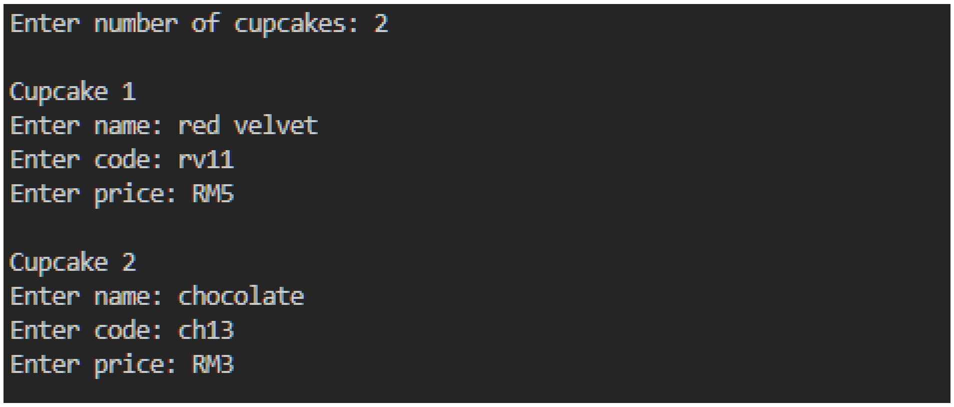


```
import java.util.Scanner;

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

        Scanner in = new Scanner(System.in);
        System.out.print("Enter number of cupcakes: ");
        int size = in.nextInt();

        // Create an array of Cupcake objects
```



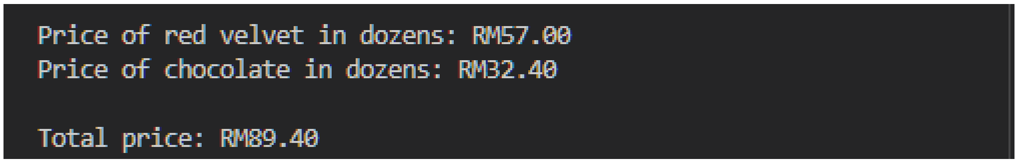
```
        Cupcake[] cupcake = new Cupcake[size];

        // Get data from user
        for (int i = 0; i < size; i++) {
            System.out.println("\nCupcake " + (i + 1));
            System.out.print("Enter name: ");
            String name = in.next();
            System.out.print("Enter code: ");
            String code = in.next();
            in.nextLine();
            System.out.print("Enter price: RM");
            double price = in.nextDouble();

            // Store data onto array
            cupcake[i] = new Cupcake(name, code, price);
        }
        in.close(); // Close scanner

        // Calculate and display price of cupcakes in dozens
        System.out.println("");
        for (int i = 0; i < size; i++) {
            System.out.printf("Price of %s in dozens: RM%.2f",
cupcake[i].getName(), cupcake[i].priceInDozen(cupcake[i].getCode()));
            System.out.println("");
        }

        // Calculate and display price of all cupcakes
        double totalPrice = 0.0;
        for (int i = 0; i < size; i++) {
            totalPrice += cupcake[i].getPrice();
        }

        // Display total price
        System.out.printf("\nTotal price: RM%.2f\n", totalPrice);
    }
}
```

SAMPLE INPUT



```
Enter number of cupcakes: 2

Cupcake 1
Enter name: red velvet
Enter code: rv11
Enter price: RM5

Cupcake 2
Enter name: chocolate
Enter code: ch13
Enter price: RM3
```

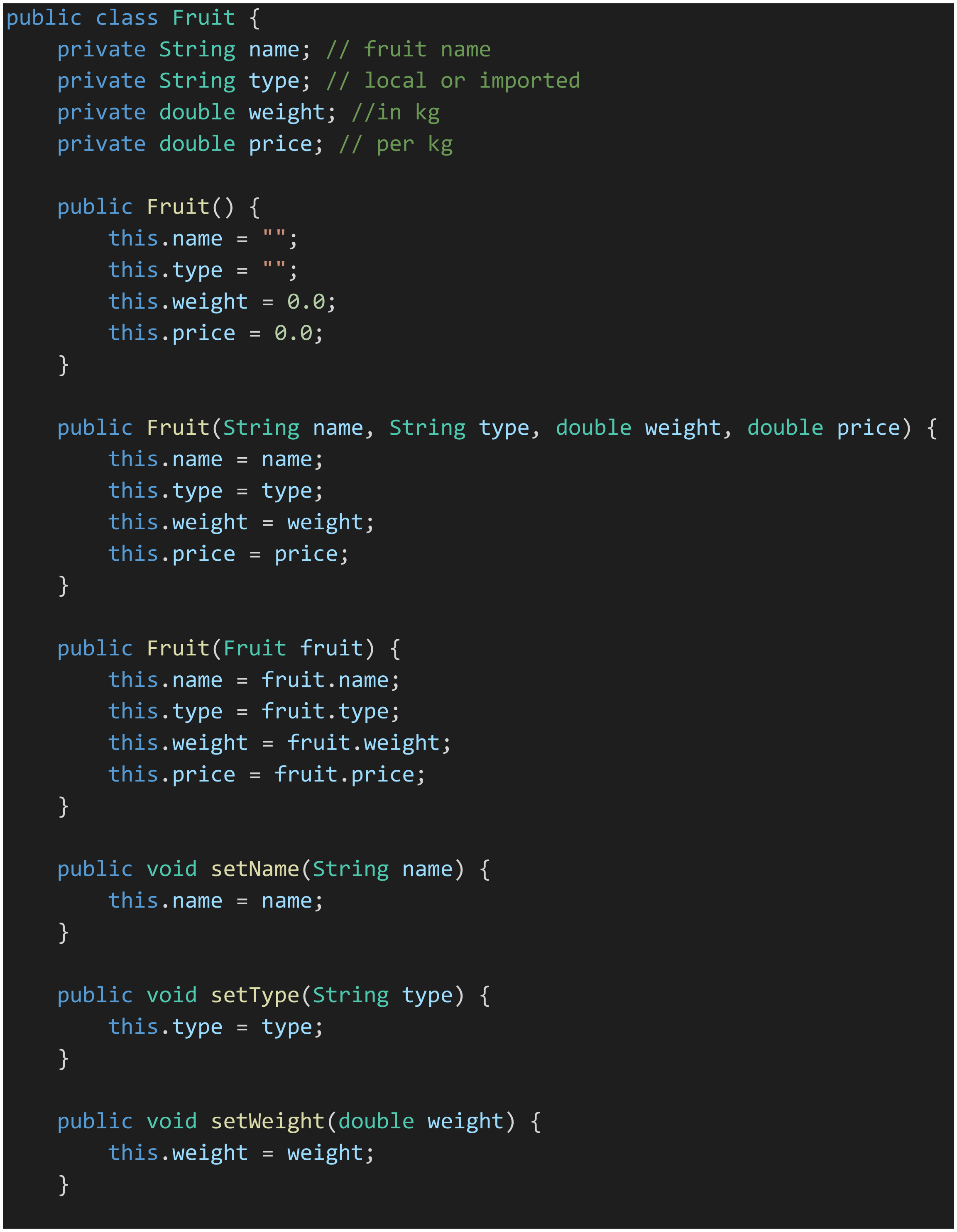
SAMPLE OUTPUT



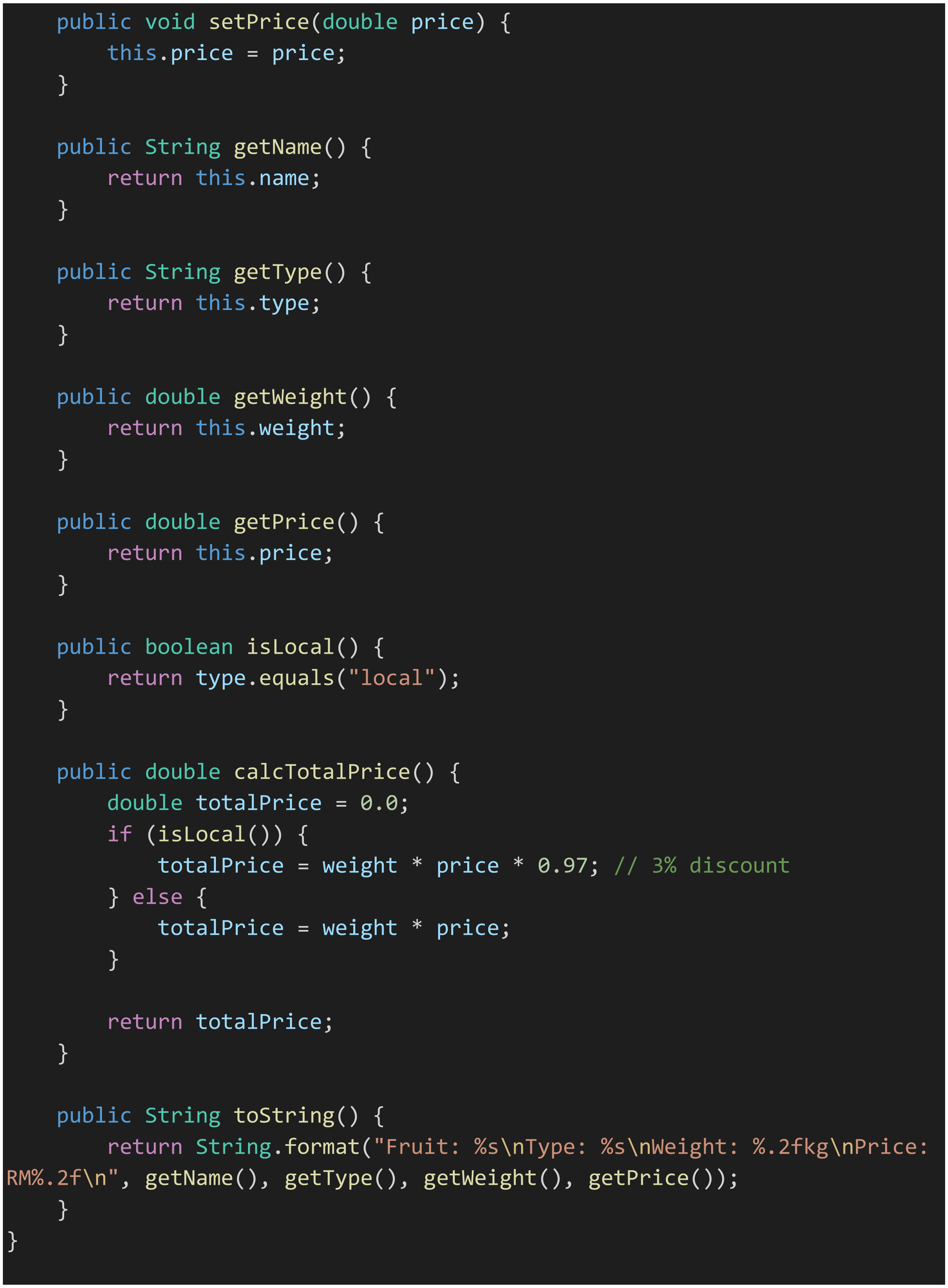
```
  Price of red velvet in dozens: RM57.00
  Price of chocolate in dozens: RM32.40

  Total price: RM89.40
```

QUESTION 3.2

SOURCE CODE 2.1 : FRUIT CLASS

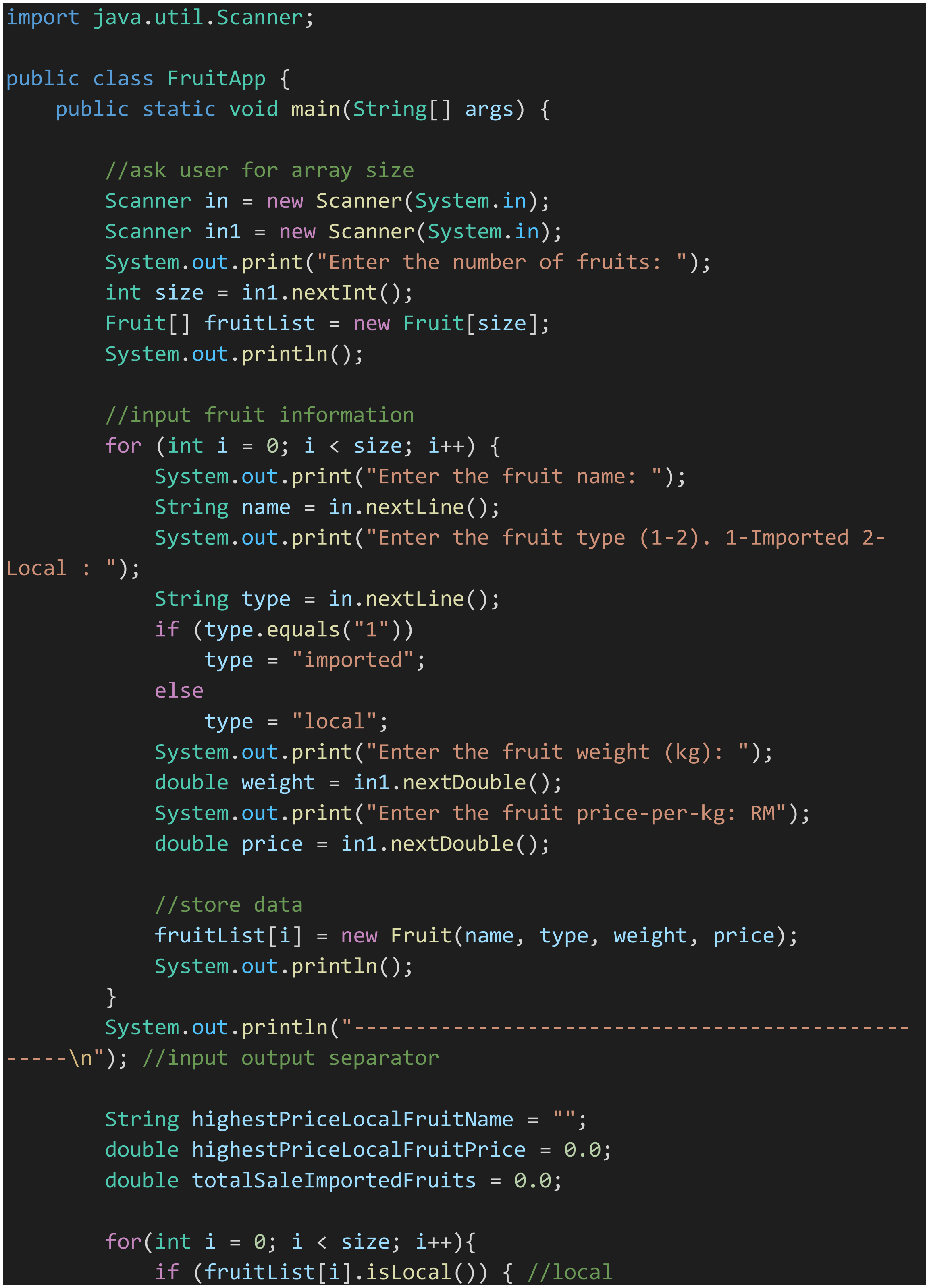


```java
public class Fruit {
    private String name; // fruit name
    private String type; // local or imported
    private double weight; //in kg
    private double price; // per kg

    public Fruit() {
        this.name = "";
        this.type = "";
        this.weight = 0.0;
        this.price = 0.0;
    }

    public Fruit(String name, String type, double weight, double price) {
        this.name = name;
        this.type = type;
        this.weight = weight;
        this.price = price;
    }

    public Fruit(Fruit fruit) {
        this.name = fruit.name;
        this.type = fruit.type;
        this.weight = fruit.weight;
        this.price = fruit.price;
    }

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

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

    public void setWeight(double weight) {
        this.weight = weight;
    }
```

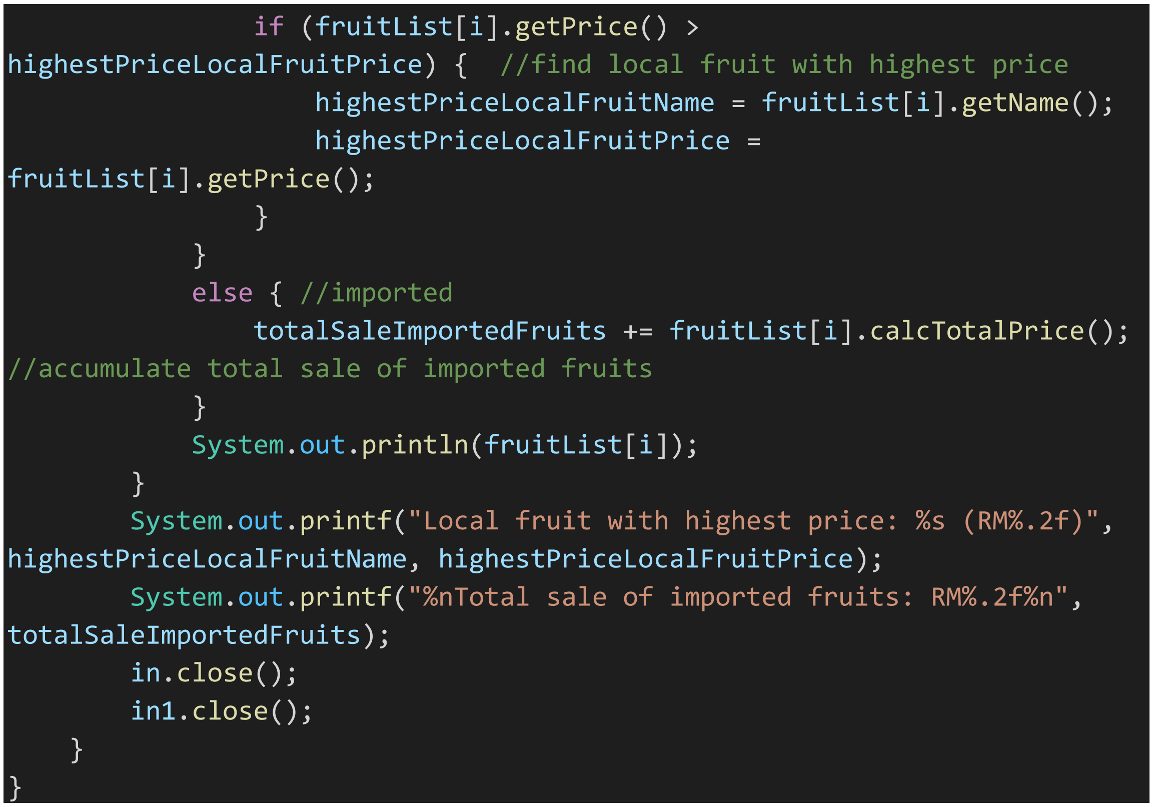


```
    public void setPrice(double price) {
        this.price = price;
    }

    public String getName() {
        return this.name;
    }

    public String getType() {
        return this.type;
    }

    public double getWeight() {
        return this.weight;
    }

    public double getPrice() {
        return this.price;
    }

    public boolean isLocal() {
        return type.equals("local");
    }

    public double calcTotalPrice() {
        double totalPrice = 0.0;
        if (isLocal()) {
            totalPrice = weight * price * 0.97; // 3% discount
        } else {
            totalPrice = weight * price;
        }

        return totalPrice;
    }

    public String toString() {
        return String.format("Fruit: %s\nType: %s\nWeight: %.2fkg\nPrice: RM%.2f\n", getName(), getType(), getWeight(), getPrice());
    }
}
```

SOURCE CODE 2.2 : FRUITAPP

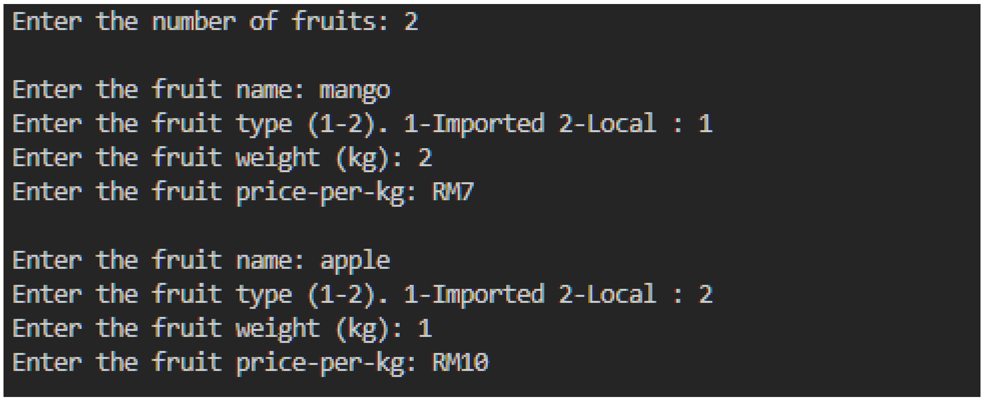


```java
import java.util.Scanner;

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

        //ask user for array size
        Scanner in = new Scanner(System.in);
        Scanner in1 = new Scanner(System.in);
        System.out.print("Enter the number of fruits: ");
        int size = in1.nextInt();
        Fruit[] fruitList = new Fruit[size];
        System.out.println();

        //input fruit information
        for (int i = 0; i < size; i++) {
            System.out.print("Enter the fruit name: ");
            String name = in.nextLine();
            System.out.print("Enter the fruit type (1-2). 1-Imported 2-Local : ");
            String type = in.nextLine();
            if (type.equals("1"))
                type = "imported";
            else
                type = "local";
            System.out.print("Enter the fruit weight (kg): ");
            double weight = in1.nextDouble();
            System.out.print("Enter the fruit price-per-kg: RM");
            double price = in1.nextDouble();

            //store data
            fruitList[i] = new Fruit(name, type, weight, price);
            System.out.println();
        }
        System.out.println("-------------------------------------------------\n"); //input output separator

        String highestPriceLocalFruitName = "";
        double highestPriceLocalFruitPrice = 0.0;
        double totalSaleImportedFruits = 0.0;

        for(int i = 0; i < size; i++){
            if (fruitList[i].isLocal()) { //local
```



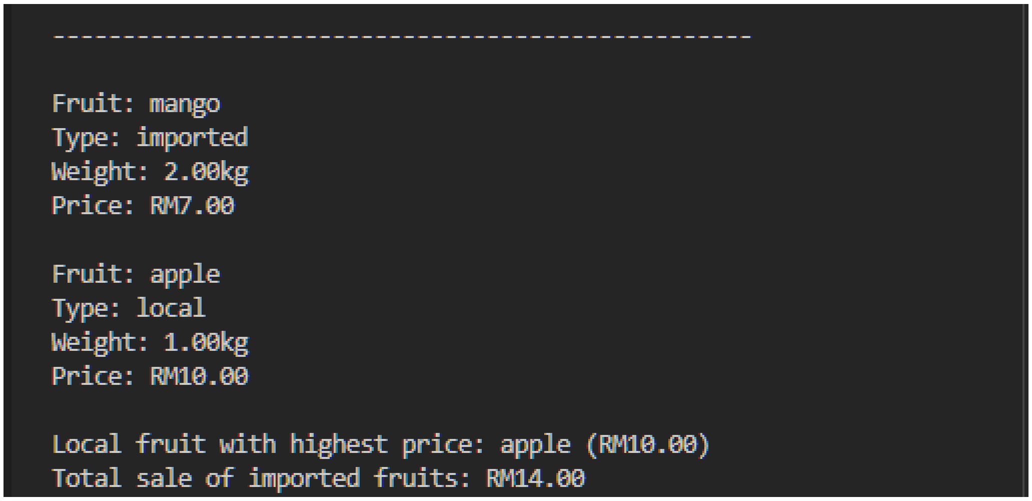
```
                if (fruitList[i].getPrice() >
highestPriceLocalFruitPrice) {  //find local fruit with highest price
                    highestPriceLocalFruitName = fruitList[i].getName();
                    highestPriceLocalFruitPrice =
fruitList[i].getPrice();
                }
            }
            else { //imported
                totalSaleImportedFruits += fruitList[i].calcTotalPrice();
//accumulate total sale of imported fruits
            }
            System.out.println(fruitList[i]);
        }
        System.out.printf("Local fruit with highest price: %s (RM%.2f)",
highestPriceLocalFruitName, highestPriceLocalFruitPrice);
        System.out.printf("%nTotal sale of imported fruits: RM%.2f%n",
totalSaleImportedFruits);
        in.close();
        in1.close();
    }
}
```

SAMPLE INPUT



```
Enter the number of fruits: 2

Enter the fruit name: mango
Enter the fruit type (1-2). 1-Imported 2-Local : 1
Enter the fruit weight (kg): 2
Enter the fruit price-per-kg: RM7

Enter the fruit name: apple
Enter the fruit type (1-2). 1-Imported 2-Local : 2
Enter the fruit weight (kg): 1
Enter the fruit price-per-kg: RM10
```

SAMPLE OUTPUT

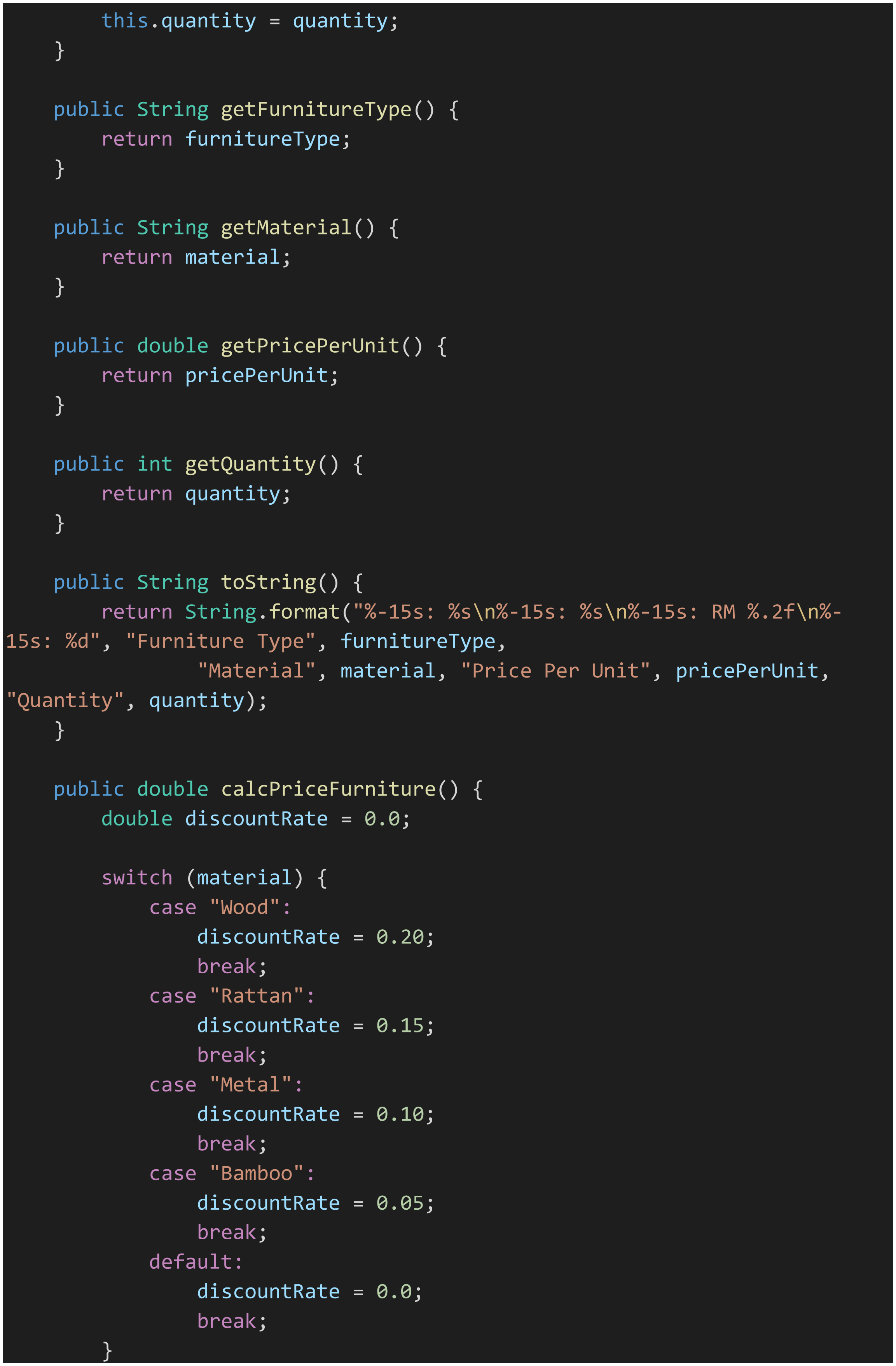


```
--------------------------------------------------

Fruit: mango
Type: imported
Weight: 2.00kg
Price: RM7.00

Fruit: apple
Type: local
Weight: 1.00kg
Price: RM10.00

Local fruit with highest price: apple (RM10.00)
Total sale of imported fruits: RM14.00
```

QUESTION 3.3

SOURCE CODE 3.1 : FURNITURE CLASS

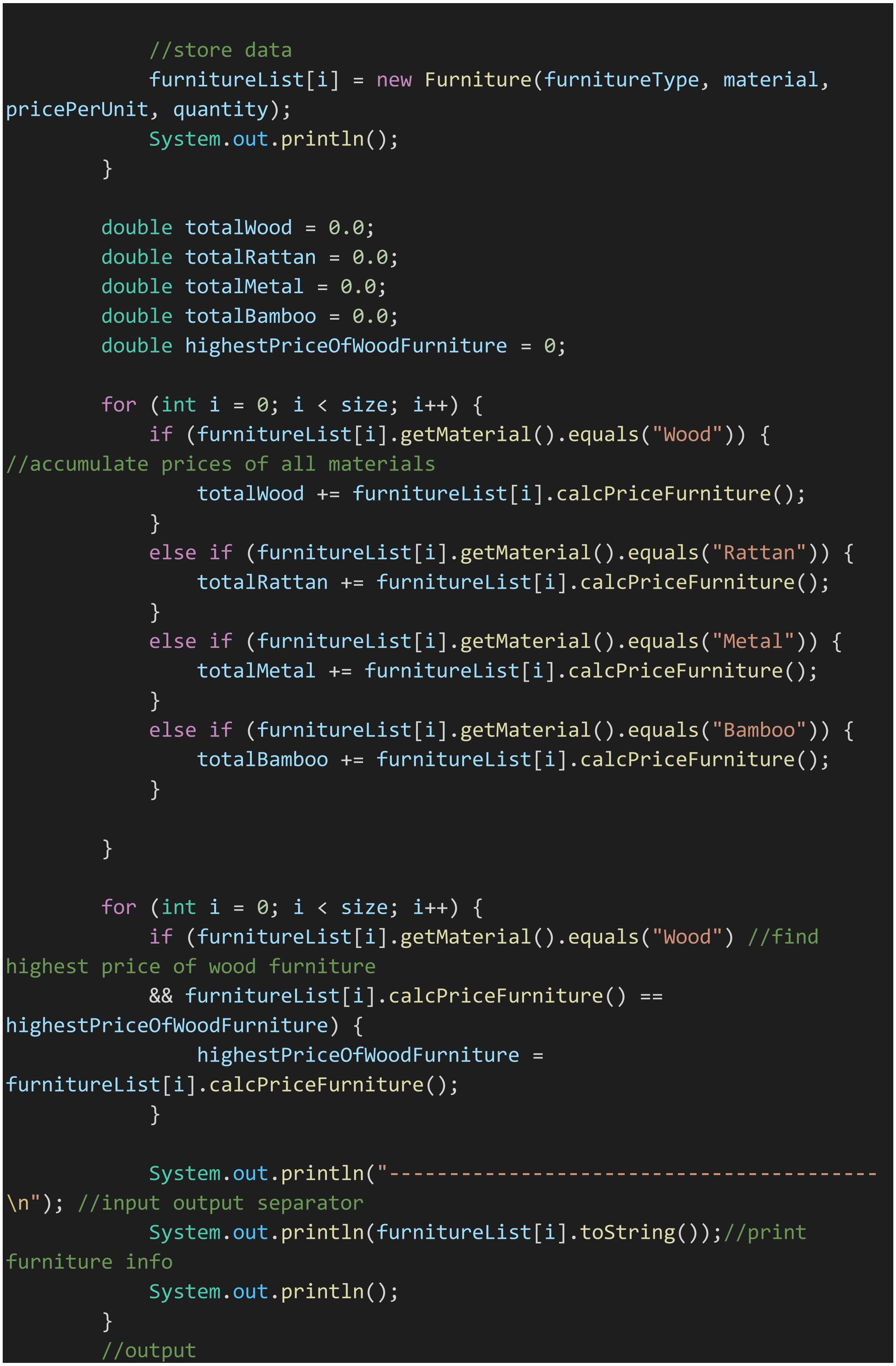
```
public class Furniture {
    private String furnitureType;
    private String material;
    private double pricePerUnit;
    private int quantity;

    public Furniture() {
        this.furnitureType = "";
        this.material = "";
        this.pricePerUnit = 0.0;
        this.quantity = 0;
    }

    public Furniture(String furnitureType, String material, double
pricePerUnit, int quantity) {
        this.furnitureType = furnitureType;
        this.material = material;
        this.pricePerUnit = pricePerUnit;
        this.quantity = quantity;
    }

    public Furniture(Furniture furniture) {
        this.furnitureType = furniture.furnitureType;
        this.material = furniture.material;
        this.pricePerUnit = furniture.pricePerUnit;
        this.quantity = furniture.quantity;
    }

    public void setFurnitureType(String furnitureType) {
        this.furnitureType = furnitureType;
    }

    public void setMaterial(String material) {
        this.material = material;
    }

    public void setPricePerUnit(double pricePerUnit) {
        this.pricePerUnit = pricePerUnit;
    }

    public void setQuantity(int quantity) {
```



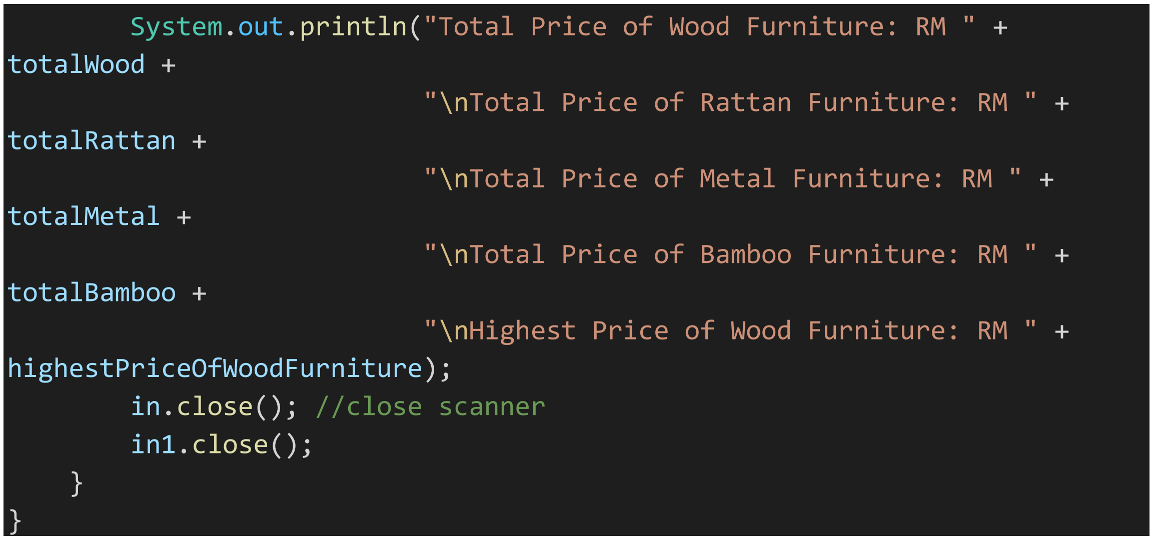
```
        this.quantity = quantity;
    }

    public String getFurnitureType() {
        return furnitureType;
    }

    public String getMaterial() {
        return material;
    }

    public double getPricePerUnit() {
        return pricePerUnit;
    }

    public int getQuantity() {
        return quantity;
    }

    public String toString() {
        return String.format("%-15s: %s\n%-15s: %s\n%-15s: RM %.2f\n%-
15s: %d", "Furniture Type", furnitureType,
                "Material", material, "Price Per Unit", pricePerUnit,
"Quantity", quantity);
    }

    public double calcPriceFurniture() {
        double discountRate = 0.0;

        switch (material) {
            case "Wood":
                discountRate = 0.20;
                break;
            case "Rattan":
                discountRate = 0.15;
                break;
            case "Metal":
                discountRate = 0.10;
                break;
            case "Bamboo":
                discountRate = 0.05;
                break;
            default:
                discountRate = 0.0;
                break;
        }
        return (pricePerUnit * quantity) * (1 - discountRate);
    }
}
```

SOURCE CODE 3.2 : FURNITUREAPP

```
import java.util.Scanner;

public class FurnitureApp {
    public static void main(String[] args) {
        //ask user for array size
        Scanner in = new Scanner(System.in);
        Scanner in1 = new Scanner(System.in);
        System.out.print("Enter the amount of furniture: ");
        int size = in1.nextInt();
        Furniture[] furnitureList = new Furniture[size];
        System.out.println();

        for (int i = 0; i < size; i++) {
            System.out.print("Enter the furniture type: ");
            String furnitureType = in.nextLine();
            System.out.print("1. Wood\n2. Rattan\n3. Metal\n4.
Bamboo\nEnter the material (1-4): ");
            String material = in.nextLine();

            switch (material) {
                case "1":
                    material = "Wood";
                    break;
                case "2":
                    material = "Rattan";
                    break;
                case "3":
                    material = "Metal";
                    break;
                case "4":
                    material = "Bamboo";
                    break;
            }
            System.out.print("Enter the price per unit: ");
            double pricePerUnit = in1.nextDouble();
            System.out.print("Enter the quantity: ");
            int quantity = in1.nextInt();
```

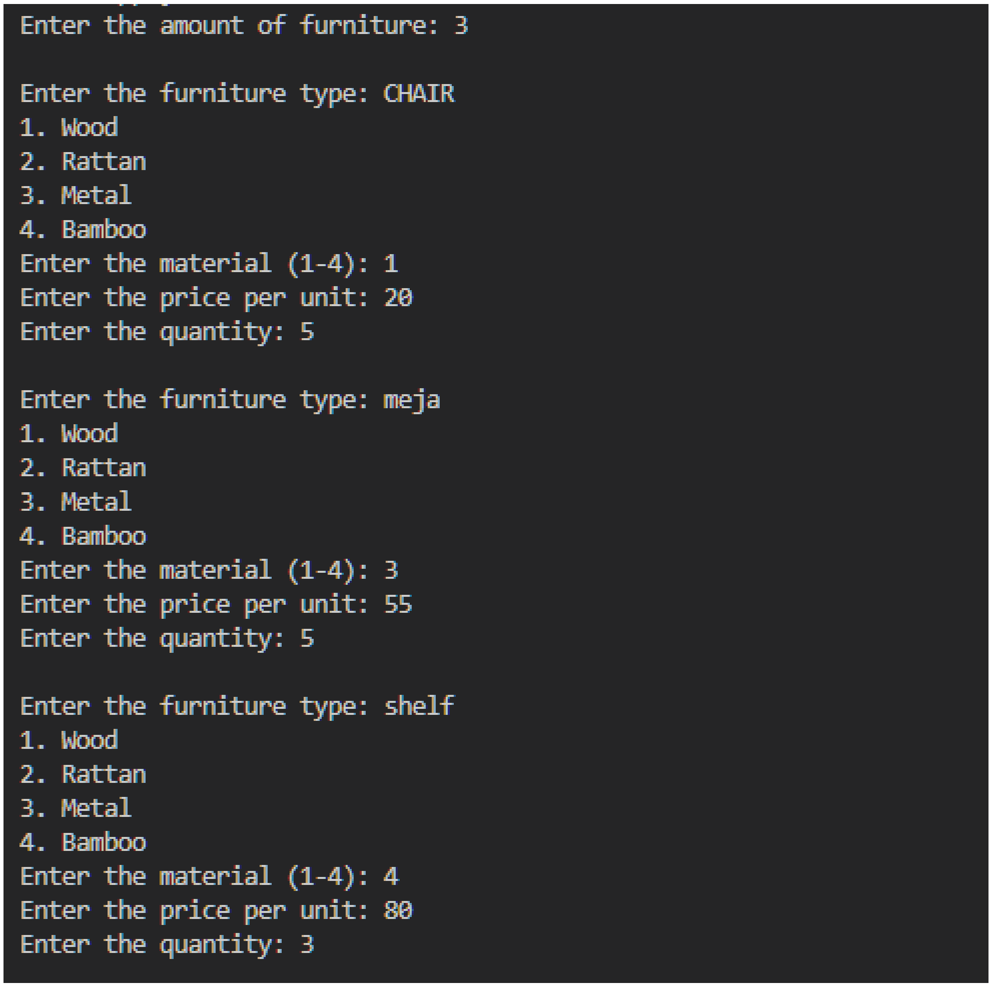


```
            //store data
            furnitureList[i] = new Furniture(furnitureType, material, 
pricePerUnit, quantity);
            System.out.println();
        }

        double totalWood = 0.0;
        double totalRattan = 0.0;
        double totalMetal = 0.0;
        double totalBamboo = 0.0;
        double highestPriceOfWoodFurniture = 0;

        for (int i = 0; i < size; i++) {
            if (furnitureList[i].getMaterial().equals("Wood")) { 
//accumulate prices of all materials
                totalWood += furnitureList[i].calcPriceFurniture();
            }
            else if (furnitureList[i].getMaterial().equals("Rattan")) {
                totalRattan += furnitureList[i].calcPriceFurniture();
            }
            else if (furnitureList[i].getMaterial().equals("Metal")) {
                totalMetal += furnitureList[i].calcPriceFurniture();
            }
            else if (furnitureList[i].getMaterial().equals("Bamboo")) {
                totalBamboo += furnitureList[i].calcPriceFurniture();
            }

        }

        for (int i = 0; i < size; i++) {
            if (furnitureList[i].getMaterial().equals("Wood") //find 
highest price of wood furniture
            && furnitureList[i].calcPriceFurniture() == 
highestPriceOfWoodFurniture) {
                highestPriceOfWoodFurniture = 
furnitureList[i].calcPriceFurniture();
            }

            System.out.println("-----------------------------------------
\n"); //input output separator
            System.out.println(furnitureList[i].toString());//print 
furniture info
            System.out.println();
        }
        //output
```



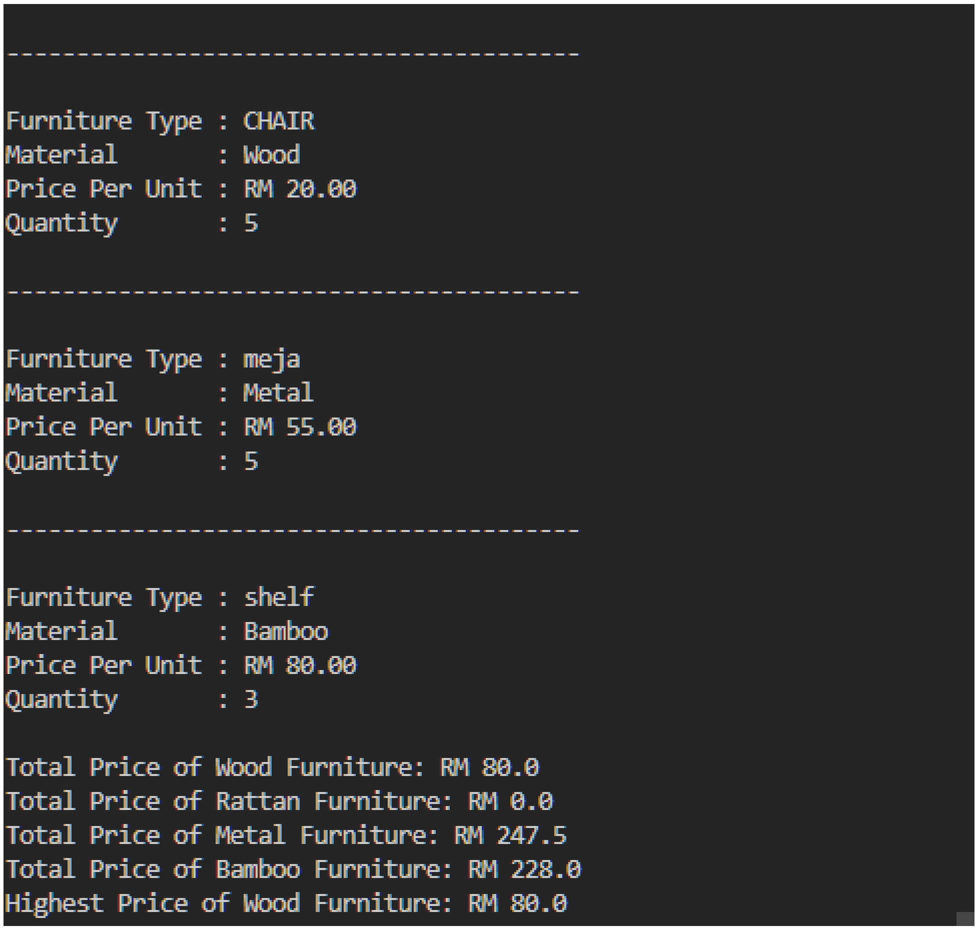
```
        System.out.println("Total Price of Wood Furniture: RM " +
totalWood +
                           "\nTotal Price of Rattan Furniture: RM " +
totalRattan +
                           "\nTotal Price of Metal Furniture: RM " +
totalMetal +
                           "\nTotal Price of Bamboo Furniture: RM " +
totalBamboo +
                           "\nHighest Price of Wood Furniture: RM " +
highestPriceOfWoodFurniture);
        in.close(); //close scanner
        in1.close();
    }
}
```

SAMPLE INPUT



```
Enter the amount of furniture: 3

Enter the furniture type: CHAIR
1. Wood
2. Rattan
3. Metal
4. Bamboo
Enter the material (1-4): 1
Enter the price per unit: 20
Enter the quantity: 5

Enter the furniture type: meja
1. Wood
2. Rattan
3. Metal
4. Bamboo
Enter the material (1-4): 3
Enter the price per unit: 55
Enter the quantity: 5

Enter the furniture type: shelf
1. Wood
2. Rattan
3. Metal
4. Bamboo
Enter the material (1-4): 4
Enter the price per unit: 80
Enter the quantity: 3
```

SAMPLE OUTPUT



```
-----------------------------------------

Furniture Type : CHAIR
Material       : Wood
Price Per Unit : RM 20.00
Quantity       : 5

-----------------------------------------

Furniture Type : meja
Material       : Metal
Price Per Unit : RM 55.00
Quantity       : 5

-----------------------------------------

Furniture Type : shelf
Material       : Bamboo
Price Per Unit : RM 80.00
Quantity       : 3

Total Price of Wood Furniture: RM 80.0
Total Price of Rattan Furniture: RM 0.0
Total Price of Metal Furniture: RM 247.5
Total Price of Bamboo Furniture: RM 228.0
Highest Price of Wood Furniture: RM 80.0
```