

# Reflection log

```
1 package Mastery;
2
3 import java.util.Scanner;
4
5 public class LunchOrder {
6
```

The code initializes the LunchOrder class in the Mastery package and imports the Scanner class to handle user input for processing lunch orders

```
7     public static void main(String[] args) {
8         // Initialize food items with their price and nutritional information
9         Food hamburger = new Food(1.85, 9, 33, 1);
10        Food salad = new Food(2.00, 1, 11, 5);
11        Food frenchFries = new Food(1.30, 11, 36, 4);
12        Food soda = new Food(0.95, 0, 38, 0);
13    }
```

This section initializes Food objects for hamburgers, salads, French fries, and sodas, each with specific price and nutritional information such as calories, fat, and protein. By defining these items, the program provides structured data to calculate total costs and nutritional details for user orders.

```
    // Create a Scanner for user input
    Scanner scanner = new Scanner(System.in);

    // Prompt user for the quantity of each item
    System.out.print("Enter the number of hamburgers: ");
    int hamburgerCount = scanner.nextInt();

    System.out.print("Enter the number of salads: ");
    int saladCount = scanner.nextInt();

    System.out.print("Enter the number of French fries: ");
    int frenchFriesCount = scanner.nextInt();

    System.out.print("Enter the number of sodas: ");
    int sodaCount = scanner.nextInt();

    // Calculate totals
    double totalPrice = (hamburger.getPrice() * hamburgerCount) +
        (salad.getPrice() * saladCount) +
        (frenchFries.getPrice() * frenchFriesCount) +
        (soda.getPrice() * sodaCount);
```

This code prompts the user to input quantities for hamburgers, salads, French fries, and sodas using a Scanner for user input. It calculates the total price by multiplying the quantity of each item by its price, retrieved using the getPrice() method, and summing

the results. This approach allows the program to dynamically calculate the cost of an order based on user input, making it suitable for simple point-of-sale applications.

```
// Separate nutritional calculationsA
int hamburgerFat = hamburger.getFat() * hamburgerCount;
int hamburgerCarbs = hamburger.getCarbs() * hamburgerCount;
int hamburgerFiber = hamburger.getFiber() * hamburgerCount;

int saladFat = salad.getFat() * saladCount;
int saladCarbs = salad.getCarbs() * saladCount;
int saladFiber = salad.getFiber() * saladCount;

int friesFat = frenchFries.getFat() * frenchFriesCount;
int friesCarbs = frenchFries.getCarbs() * frenchFriesCount;
int friesFiber = frenchFries.getFiber() * frenchFriesCount;

int sodaFat = soda.getFat() * sodaCount;
int sodaCarbs = soda.getCarbs() * sodaCount;
int sodaFiber = soda.getFiber() * sodaCount;
```

This code section calculates the nutritional content for various food items by multiplying the item's nutritional values (fat, carbohydrates, and fiber) by their respective quantities. It defines separate variables for each nutrient type and food item, ensuring the results are stored in a structured and readable way. This approach allows the program to aggregate and use these values in subsequent calculations, such as generating a total nutritional summary.

```
// Display the order summary for each item
System.out.println("\nOrder Summary:");

System.out.println("\nHamburgers:");
System.out.println("  Fat: " + hamburgerFat + "g");
System.out.println("  Carbs: " + hamburgerCarbs + "g");
System.out.println("  Fiber: " + hamburgerFiber + "g");

System.out.println("\nSalads:");
System.out.println("  Fat: " + saladFat + "g");
System.out.println("  Carbs: " + saladCarbs + "g");
System.out.println("  Fiber: " + saladFiber + "g");

System.out.println("\nFrench Fries:");
System.out.println("  Fat: " + friesFat + "g");
System.out.println("  Carbs: " + friesCarbs + "g");
System.out.println("  Fiber: " + friesFiber + "g");

System.out.println("\nSodas:");
System.out.println("  Fat: " + sodaFat + "g");
System.out.println("  Carbs: " + sodaCarbs + "g");
System.out.println("  Fiber: " + sodaFiber + "g");

// Display totals
System.out.printf("\nTotal price: $%.2f\n", totalPrice);
System.out.println("Total fat: " + (hamburgerFat + saladFat + friesFat + sodaFat) + "g");
System.out.println("Total carbohydrates: " + (hamburgerCarbs + saladCarbs + friesCarbs + sodaCarbs) + "g");
System.out.println("Total fiber: " + (hamburgerFiber + saladFiber + friesFiber + sodaFiber) + "g");

// Close the scanner
scanner.close();
```

**This section of the code displays a detailed order summary for each food item, including the amounts of fat, carbohydrates, and fiber in grams. It also calculates and outputs the total price, total fat, total carbohydrates, and total fiber of the order. The code ensures clarity in the nutritional breakdown and closes the scanner after completing the output, signaling the end of input handling.**