Assignment name: LunchOrder Mastery Project - Chapter 7
Student name: Misha Stanev

Reflection Log

Planning: Idea during planning was to put TestLunchOrder and LunchOrder into one file.
Coding: I put them into two separate files.
Now: Same as coding.

TestLunchOrder

Create a class named TestLunchOrder with the main method inside it. Within this method I turned LunchOrder into a useable class named Order, and I formatted the decimal. (Shown below)

```
import java.text.DecimalFormat;
import java.util.*;

public class TestLunchOrder {
    public static void main(String[] args) {
        Scanner userinput = new Scanner(System.in);
        LunchOrder order = new LunchOrder();
        DecimalFormat df = new DecimalFormat("0.00");
```

Print menu to user. Menu contains price, and the nutritional facts of each item (Shown below)

```
System.out.println("Item          Price     Fat(g)     Carbohydrates(g)     Fiber(g)");
System.out.println("Hamburger     $1.85       9              33                1");
System.out.println("Salad         $2.00       1              11                5");
System.out.println("French Fries  $1.30       11             36                4");
System.out.println("Soda          $0.95       0              38                0");
```

Prompt's user for amount of hamburgers, salads, fries, and sodas, and show the nutritional value for each item (Shown below)

```
System.out.print("Enter amount of hamburgers: ");
order.setHamburger(userinput.nextInt());
System.out.println("Each Hamburger has 9g of fat, 33g of carbs, and 1g of fiber.");
System.out.println("  ");

System.out.print("Enter number of salads: ");
order.setSalad(userinput.nextInt());
System.out.println("Each salad has 1g of fat, 11g of carbs, and 5g of fiber.");
System.out.println("  ");

System.out.print("Enter number of French Fries: ");
order.setFries(userinput.nextInt());
System.out.println("Each french fries has 11g of fat, 36g of carbs, and 0g of fiber.");
System.out.println("  ");

System.out.print("Enter number of soda: ");
order.setSoda(userinput.nextInt());
System.out.println("Each soda has 0g of fat, 38g of carbs, and 0g of fiber.");
System.out.println("  ");

System.out.print("Your total is: $" + df.format(order.getTotal()));
```

LunchOrder

Create class and variables (Shown below)

```
public class LunchOrder {
    private double total;  // Total cost of the lunch order
    private int hamburger, salad, fries, soda;  // Amount of each item in the order
```

Method to calculate the total cost of the order depending on the price of each item (Shown below)

```
public void setLunchOrder() {
    // The total is calculated by multiplying the quantity of each item by its price
    total = hamburger * 1.85 + salad * 2.00 + fries * 1.30 + soda * 0.95;
}
```

Method that adds number of hamburgers to total, then recalculates total (Shown below)

```
public void setHamburger(int h) {
    hamburger += h;  // Add the number of hamburgers to the current total
    setLunchOrder();  // Recalculate the total
```

Method that adds number of salads to total, then recalculates the total (Shown below)

```
public void setFries(int f) {
    fries += f;  // Add the number of fries to the current total
    setLunchOrder();  // Recalculate the total after updating the quantity
```

Method that adds number of fries to total, then recalculates total (Shown below)

```
public void setFries(int f) {
    fries += f;  // Add the number of fries to the current total
    setLunchOrder();  // Recalculate the total after updating the quantity
```

Method that adds number of sodas to total, then recalculates total (Shown below)

```
public void setSoda(int S) {
    soda += S;  // Add the number of sodas to the current total
    setLunchOrder();  // Recalculate the total after updating the quantity
}
```

Method to calculate total cost of the order, then returns that number (Shown below)

```
public double getTotal() {
    return total;  // Return the current total cost of the order
}
```