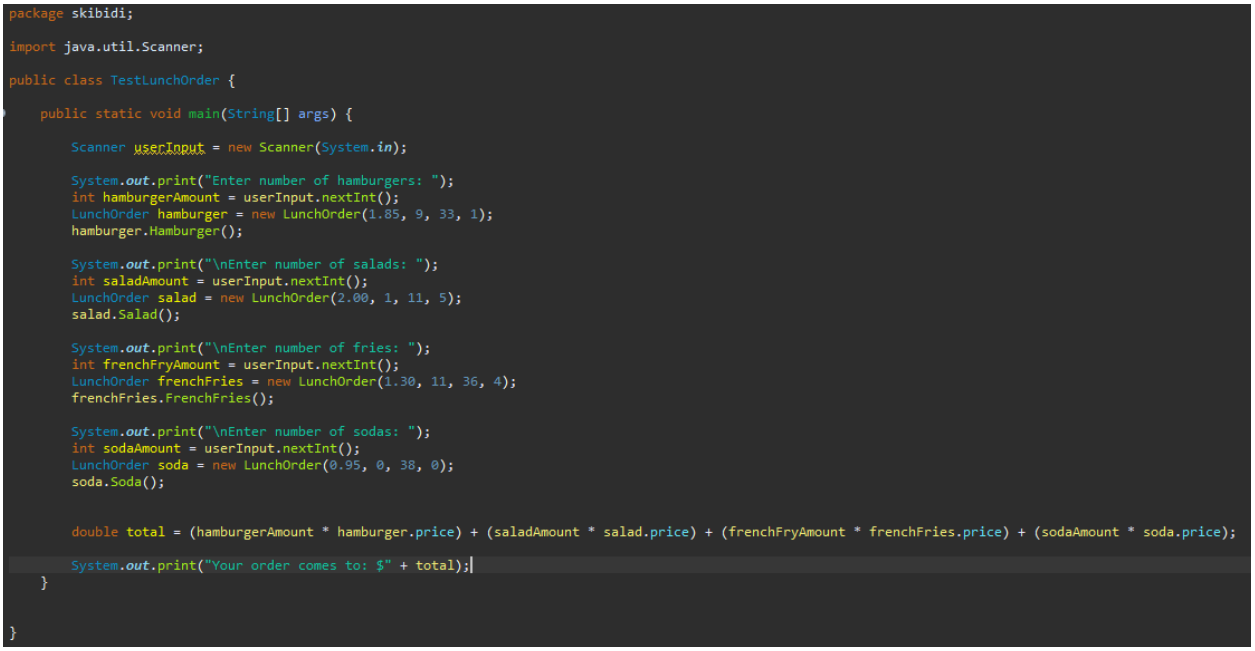
DigitExtractor Mastery ErrorLog

Foreword: Screenshots were taken on both my home and school computer, hence the difference between the package in some of the screenshots.

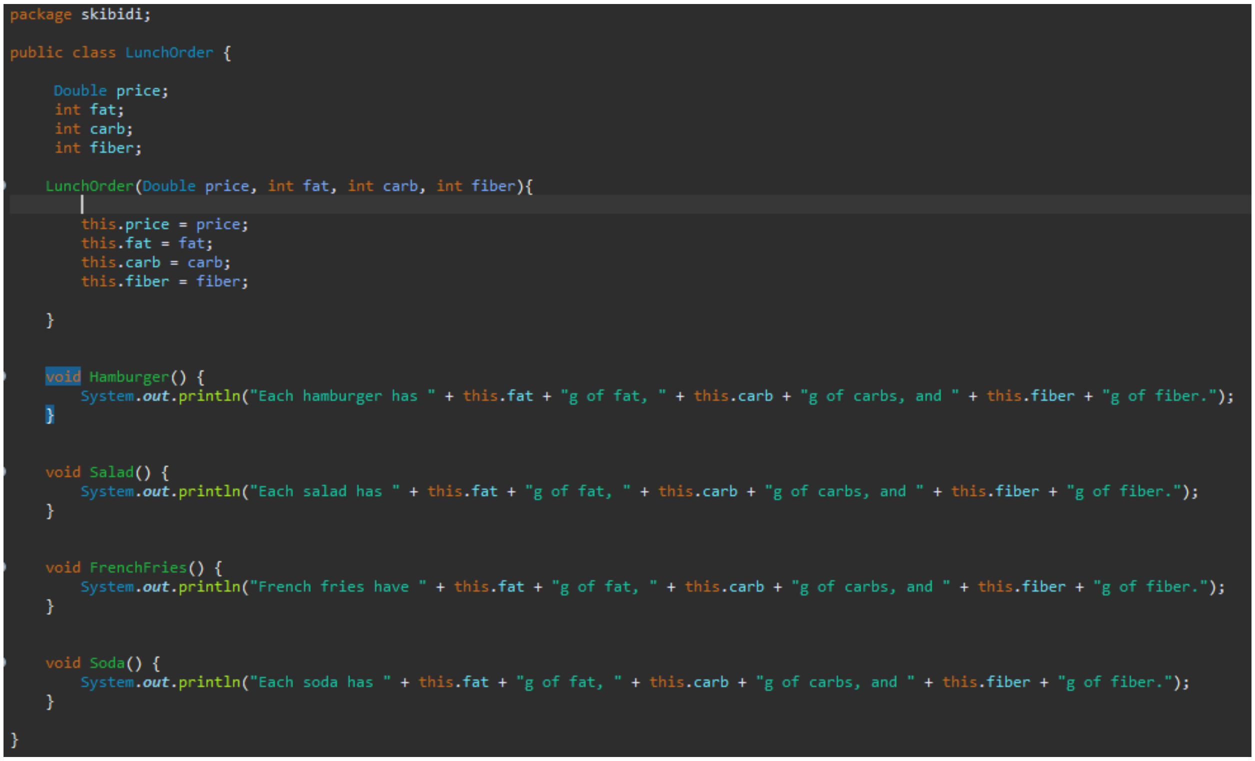
Below is the first rendition of the TestLunchOrder class:



```java
package skibidi;

import java.util.Scanner;

public class TestLunchOrder {

    public static void main(String[] args) {

        Scanner userInput = new Scanner(System.in);

        System.out.print("Enter number of hamburgers: ");
        int hamburgerAmount = userInput.nextInt();
        LunchOrder hamburger = new LunchOrder(1.85, 9, 33, 1);
        hamburger.Hamburger();

        System.out.print("\nEnter number of salads: ");
        int saladAmount = userInput.nextInt();
        LunchOrder salad = new LunchOrder(2.00, 1, 11, 5);
        salad.Salad();

        System.out.print("\nEnter number of fries: ");
        int frenchFryAmount = userInput.nextInt();
        LunchOrder frenchFries = new LunchOrder(1.30, 11, 36, 4);
        frenchFries.FrenchFries();

        System.out.print("\nEnter number of sodas: ");
        int sodaAmount = userInput.nextInt();
        LunchOrder soda = new LunchOrder(0.95, 0, 38, 0);
        soda.Soda();


        double total = (hamburgerAmount * hamburger.price) + (saladAmount * salad.price) + (frenchFryAmount * frenchFries.price) + (sodaAmount * soda.price);

        System.out.print("Your order comes to: $" + total);
    }


}
```

Within my first rendition of the code the price would sometimes create decimal errors during the price calculation stage.

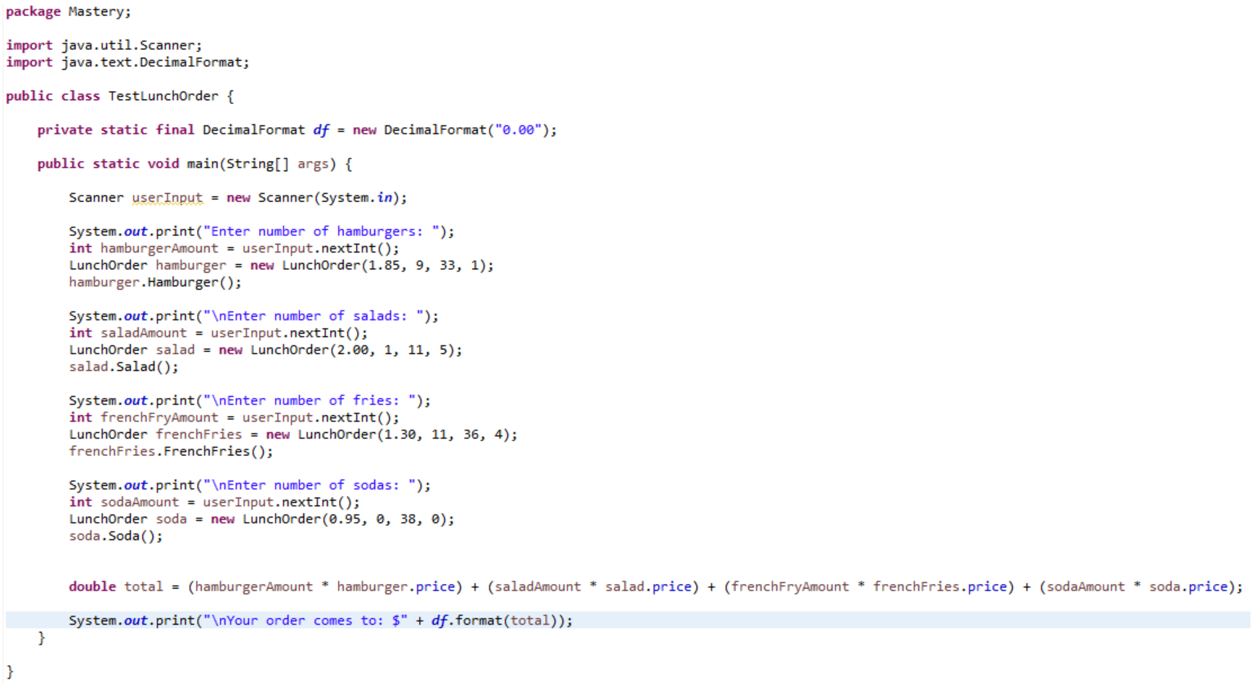
Below is the first rendition of the LunchOrder class:



```
package skibidi;

public class LunchOrder {

     Double price;
     int fat;
     int carb;
     int fiber;

    LunchOrder(Double price, int fat, int carb, int fiber){

        this.price = price;
        this.fat = fat;
        this.carb = carb;
        this.fiber = fiber;

    }


    void Hamburger() {
        System.out.println("Each hamburger has " + this.fat + "g of fat, " + this.carb + "g of carbs, and " + this.fiber + "g of fiber.");
    }


    void Salad() {
        System.out.println("Each salad has " + this.fat + "g of fat, " + this.carb + "g of carbs, and " + this.fiber + "g of fiber.");
    }


    void FrenchFries() {
        System.out.println("French fries have " + this.fat + "g of fat, " + this.carb + "g of carbs, and " + this.fiber + "g of fiber.");
    }


    void Soda() {
        System.out.println("Each soda has " + this.fat + "g of fat, " + this.carb + "g of carbs, and " + this.fiber + "g of fiber.");
    }

}
```

No errors were present in the LunchOrder class.

Below is the final rendition of the TestLunchOrder class:



```
package Mastery;

import java.util.Scanner;
import java.text.DecimalFormat;

public class TestLunchOrder {

    private static final DecimalFormat df = new DecimalFormat("0.00");

    public static void main(String[] args) {

        Scanner userInput = new Scanner(System.in);

        System.out.print("Enter number of hamburgers: ");
        int hamburgerAmount = userInput.nextInt();
        LunchOrder hamburger = new LunchOrder(1.85, 9, 33, 1);
        hamburger.Hamburger();

        System.out.print("\nEnter number of salads: ");
        int saladAmount = userInput.nextInt();
        LunchOrder salad = new LunchOrder(2.00, 1, 11, 5);
        salad.Salad();

        System.out.print("\nEnter number of fries: ");
        int frenchFryAmount = userInput.nextInt();
        LunchOrder frenchFries = new LunchOrder(1.30, 11, 36, 4);
        frenchFries.FrenchFries();

        System.out.print("\nEnter number of sodas: ");
        int sodaAmount = userInput.nextInt();
        LunchOrder soda = new LunchOrder(0.95, 0, 38, 0);
        soda.Soda();


        double total = (hamburgerAmount * hamburger.price) + (saladAmount * salad.price) + (frenchFryAmount * frenchFries.price) + (sodaAmount * soda.price);

        System.out.print("\nYour order comes to: $" + df.format(total));
    }

}
```

To fix the rounding errors I included a df.Format function to round to 2 decimal places. No other errors were present.