

## CSC 151 Project 1 – TakeOut project – Phase 2

Continue with all requirements from the original program with the following changes. All requirements remain, with many of the changes highlighted. Please read ***entire document carefully***.

### Use Methods

Modify your first phase of this project to incorporate the following methods:

```
public static void displayMainMenu()  
public static void displayItalianFoodMenu()  
public static void displayChineseFoodMenu()  
public static void displayAmericanFoodMenu()  
public static int determineTrays(int people, int feeds)  
public static double getSubtotal(double price, int trays)  
public static double getTax(double subtotal, double taxrate)  
public static double getTip(double subtotal, double tipRate)  
public static double getGrandTotal(double subtotal, double tax, double tip)  
public static double pricePerPerson(double grandTotal, int people)  
public static int determineLeftOvers(int feeds, int trays, int people)
```

**\*\* These method signatures must implemented as shown, using the same return type, name and parameter list.**

**\*\* These methods should be inside your Class, but outside of the main method, underneath the main method. Remember, you cannot nest methods, or have a method within a method.**

**\*\* All methods should be called from the main method. Do not have methods (other than the main) call other methods**

### Use Looping

Modify your first phase of this program to continue to ask the user for a correct menu choice if they enter a choice that is not valid, using a while or do while loop. The loop should NOT be in the display menu methods as they are void (non returning methods).

- In order for your work to be graded, the project needs to be named with yourlastnameTakeOutProject (e.g. HalseyTakeOutProject), and your .java class file needs to be named yourlastnameTakeOut (e.g. HalseyTakeOut).
- You must zip your project folder and submit that folder to Moodle by the due date.
- Projects must be named and submitted to Moodle correctly.

Premise:

A group of tenants from an apartment complex have decided to band together and do a group take out (to be delivered) during quarantine.

They have three restaurants that they choose from, an Italian, American and Chinese food place. Each restaurant sells three options of family style trays.

The number of people placing an order may vary by each time. One day only 11 might order, one day maybe 24. There are 100 people in the apartment complex and they have decided that they need at least 10 people to justify a group order.

Your program will determine how many trays of food they will need, the 7% tax, a 15% tip and a total price and then price per person.

There may be left over servings, the group has decided to donate any leftover servings to the delivery person.

The program flow will be:

1. Ask how many people are ordering, if the number is out of range, display a message that number of people are invalid and stop the program (do not use loops)
2. Ask what type of food will be ordered, using the following menu. If the number or character entered is invalid, display a message and stop the program (do not use loops):

CHOOSE TYPE OF MEAL

- 
1. Italian
  2. Chinese
  3. American

Choose what type of food you'd like to order:

3. Based on the type of food chosen, display appropriate menu and ask which food tray to order. If the number or character entered is invalid, display a message and stop the program (do not use loops)::

CHOOSE ONE

- 
1. Lasagna Tray - Feeds 5 - 17.99
  2. Pizza Pack - Feeds 7 - 15.99
  3. Gazpacho Soup, salad and bread sticks pack - Feeds 4 - 12.99
- Enter Choice

CHOOSE ONE

1. Chicken and Broccoli Tray (includes 7 wonton soups, 7 egg rolls) - feeds 7 - 18.99
2. Sweet and Sour Pork Tray (includes 7 hot and sour soups, 7 egg rolls) - feeds 7 - 18.99
3. Shrimp Fried Rice Tray (includes 10 egg rolls) - Feeds 5 - 10.99

Enter Choice

#### CHOOSE ONE

1. Hamburger and Hot Dog Tray - includes buns and condiments - feeds 8 - 21.99
2. Grilled Chicken Sandwich and Mozzarella Sticks Tray - includes dipping sauces -Feeds 5 - 22.99
3. Barbeque Tray - includes buns and peach cobbler - Feeds 10 - 26.99

Enter Choice

4. Determine how many trays
5. Determine price before tax
6. Determine tax (based on total price before tax or tip)
7. Determine tip (based on total price before tax or tip)
8. Determine total (price + tax + tip); use currency computation
9. Determine price per person (based on total – price + tax + tip)
10. Determine how many leftover servings go to the delivery person
11. Output your results (see below example):

You need 3 trays  
 Feeds: 5  
 Price for 12 people (3 trays): 53.97  
 Tax: 3.78  
 Tip: 8.10  
 Total (food, tax, tip): 65.85  
 Price per person: 5.49  
 Leftover servings for the delivery person:3

Item	Points	
<p>Project and class named and submitted correctly          Academic integrity statement, name, date,          purpose at top of program</p> <p>Example:          // Rebecca Halsey 5/16/2020          // Take out program project</p>		<p>Required for grading –</p> <p>Projects not submitted correctly or missing the academic integrity statement will not be graded.</p>

<pre>// CSC 151 MON01 // I abide by GTCC's academic integrity policy // this work is my own original work</pre>		
<p>Accurately completing the above specified project using correct and efficient use of the following elements:</p> <ul style="list-style-type: none"> <li>• Variables and data types</li> <li>• System output (System.out.print(ln)</li> <li>• Java Scanner class, input</li> <li>• Comparison operators</li> <li>• Mathematical expressions and computations</li> <li>• Currency computations</li> <li>• At least one switch statement</li> <li>• At least one if/else structure</li> <li>• All methods correctly written, called and return values (if applicable) used correctly</li> <li>• Looping for data validation implemented and functioning correctly</li> <li>• Method body code programmed accurately and efficiently</li> <li>• method signatures must implemented as shown, using the same return type, name and parameter list.</li> <li>• All methods should be called from the main method. Do not have methods (other than the main) call other methods</li> </ul>	80	
<ul style="list-style-type: none"> <li>• Professional and clear program prompts</li> <li>• Professional program flow</li> <li>• Professional comments within your code</li> <li>• Professional output formatting</li> <li>• Professional program style, convention, indentation, variable names, etc.</li> <li>• Format currency amounts showing two decimal points... example: 2.10 , 3.45, 50.20, 60.00, etc.</li> </ul>	20	

Total Points	100	
Extra Credit:	10	Write a method to display the totals.