

CS 1313 010: Programming for Non-majors, Fall 2014
Programming Project #4: Italian Restaurant
Due by 11:20am Wednesday October 22 2014

This fourth project will give you experience writing programs that involve branching (*if*). This project will use the same development process as in Programming Projects #2 & #3, and will be subject to the same rules and grading criteria, with some new criteria added. This specification is less detailed than for previous programming projects. **YOU ARE EXPECTED TO KNOW HOW TO PERFORM BASIC TASKS WITHOUT HAVING TO BE TOLD EXPLICITLY.**

You've just opened a Italian restaurant.

Your Italian restaurant has a menu that features several appetizers (antipasto, bruschetta, minestrone soup), several entrees (eggplant parmesan, fettucini, gnocchi, ravioli, rigatoni, vegetarian lasagna), and several drinks (coffee, soda, tea).

Each meal can consist of an appetizer, an entree and a drink.

Each entree has two size options: small and large. (Appetizers and drinks come in one size only.)

Entree prices depend on the size choice: the small portion is cheaper than the large portion for each entree choice. By contrast, appetizer prices and drink prices **DON'T** have size options.

A patron can order **AT MOST** one item from each category; for example, they can order antipasto as their appetizer, a large vegetarian lasagna as their entree, and tea as their drink. But, they **AREN'T REQUIRED** to order one of each category; for example, they can order no appetizer or entree at all, just a cup of coffee.

The prices are:

Item	Price	Item	Small Price	Large Price
Antipasto	\$8.00	Eggplant Parmesan	\$11.00	\$14.50
Bruschetta	\$5.75	Fettucini	\$14.00	\$16.75
Minestrone Soup	\$6.25	Gnocchi	\$13.75	\$19.50
Coffee	\$4.25	Ravioli	\$11.75	\$14.00
Soda	\$2.00	Rigatoni	\$16.00	\$18.75
Tea	\$3.50	Vegetarian Lasagna	\$17.75	\$20.00

Your community's sales tax rate is 8.25%.

NOTE: You are welcome to substitute other items, as long as you do so consistently and continue to follow the rules. For example, you could change fettucini to angel hair pasta, eggplant parmesan to chicken parmesan, etc, but the number and categories of items, and the prices, must remain the same.

I. PROJECT DESCRIPTION

Write a program that simulates greeting a customer, taking their order, calculating the bill and producing the check. The computer will be the wait staff, and the user will be the customer.

EXTRA PREPROCESSOR DIRECTIVE

Your program **MUST** start with the following **TWO** preprocessor directives, in this order:

```
#include <stdio.h>
#include <stdlib.h>
```

II. STRUCTURE OF THE PROGRAM

The program body **MUST** be broken into four subsections:

1. **Greeting subsection:** Greet the user.
2. **Input subsection:** Input their order, one item at a time, **IDIOTPROOFING EACH VALUE AS SOON AS IT IS INPUT.**
3. **Calculation subsection:** Determine the price of each item, the subtotal, the tax amount, and the grand total.
4. **Output subsection:** Output their bill in receipt form.

Please note that you are **ABSOLUTELY FORBIDDEN** to have:

- **ANY** executable statements in your declaration section;
- **ANY** declarations in your execution section (body);
- **ANY** inputs or calculations in your greeting subsection;
- **ANY** calculations, or outputs other than prompts and idiotproofing error messages, in your input subsection;
- **ANY** inputs or outputs in your calculation subsection;
- **ANY** inputs or calculations in your output subsection.

That is, the subsections **MUST BE COMPLETELY SEPARATE,** and **MUST BE CLEARLY LABELED.**

For purposes of this programming project, `if` blocks are not considered to be inputs, nor calculations, nor outputs; that is, in principle you may have an `if` block in **ANY** subsection of the program body. However, statements **inside** the clauses of an `if` block **MUST** follow the rules above.

III. IMPLEMENTATION ORDER

Because the program will be complicated, you are **STRONGLY** advised to **IMPLEMENT ONE PART AT A TIME,** thoroughly test and debug it, and then go on to the next part. Also, it would probably be best to implement the subsections, **NOT** in the order in which they appear in the program (as described above), but rather in the following order:

1. **Greeting subsection.**
2. **Input subsection** (developed one item at a time). **NOTE:** If you're unclear on how to idiotproof, then you can skip the idiotproofing during initial implementation and then develop the idiotproofing code later.
3. **Output subsection** (developed one item at a time). Note that, at this stage, some or all of your outputs will be garbage, because you haven't yet written the calculation subsection.
4. **Calculation subsection** (developed one item at a time).

IV. DETAILS OF THE PROGRAM STRUCTURE

A. Greet the Customer

Welcome the customer to the restaurant.

B. Input the Customer's Order

1. Ask (prompt) the customer for their appetizer item choice. You can use integer-valued codes to represent the appetizer item choices, and you may choose any REASONABLE values for coding these items. So, when the program prompts them for their appetizer item choice, it might OUTPUT something like:

```
What appetizer item would you like?
```

```
Please enter:
```

```
0  for no appetizer
1  for antipasto
2  for bruschetta
3  for minestrone soup
```

2. Input the customer's appetizer item choice.
3. IDIOTPROOF the appetizer item choice, to ensure that the value that the user has input is one of the values listed in the prompt. (See the Grading Criteria for details.)
4. Ask (prompt) the customer for their entree item choice, giving them a list of entree items to choose from. You can use integer-valued codes to represent the entree items, and you may choose any REASONABLE values for coding these items. So, when the program prompts them for their entree item choice, it might OUTPUT something like:

```
What entree item would you like?
```

```
Please enter:
```

```
0  for no entree
1  for eggplant parmesan
2  for fettucini
3  for gnocchi
4  for ravioli
5  for rigatoni
6  for vegetarian lasagna
```

5. Input the customer's entree item choice.
6. IDIOTPROOF the entree item choice, to ensure that the value that the user has input is one of the values listed in the prompt. (See the Grading Criteria for details.)