

Restricted Restaurant Menu Design Document

Sushil J. Louis
sushil@cse.unr.edu
<http://www.cse.unr.edu/~sushil>

Abstract

This program implements a restricted restaurant menu and billing system for the Virtual Street Cafe. I primarily use the switch statement to implement the menu system and to ask the customer for choices from a menu tree. File I/O functions print a bill that includes taxes and tip to standard output and to a file. The program obtains information on available appetizers, entrees, and desserts from text files containing the names and prices of these menu items. Item names are restricted to single words and the program does not handle multi-word (whitespace separated) menu items. Furthermore, to simplify implementation, I make the assumption that each menu contains exactly three (3) items.

1 Design Logic

Figure 1 shows an example tree menu for the Virtual Cafe.

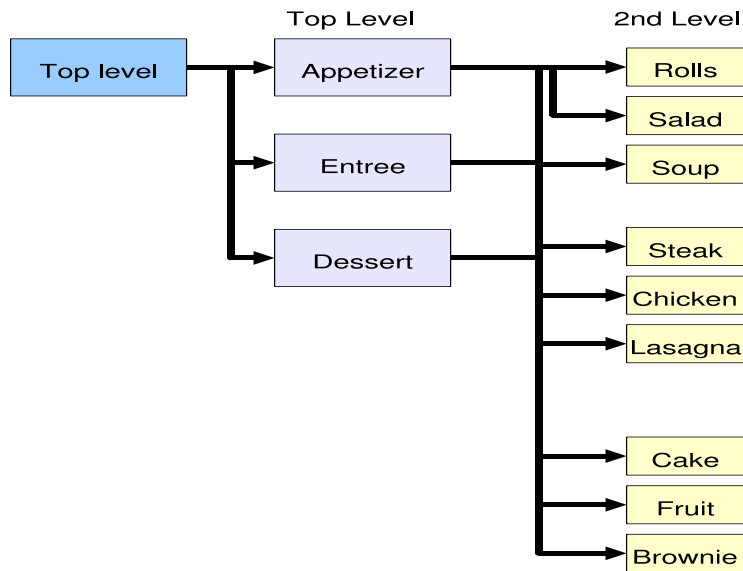


Figure 1: A sample restaurant menu tree

The program works in three stages. First, it uses the function `displayChoices` to display the top level menu consisting of the following three possibilities.

1. Appetizer
2. Entree
3. Dessert

It then reads in the user's choice (the integer 1, 2 or 3) and uses a switch statement to decide the parameters for the function `selectFile` which then reads items and prices from the appropriate file and displays second level choices to the user. The program exits with an error message if the user makes an invalid first level choice.

Second, once the user makes the second level choice the program uses the price for this choice to calculate taxes as $0.073750 \times$ the price and the tip as $0.15 \times$ the price. If the user makes an invalid choice at either level, the program prints an error message and exits. Note that `selectFile` returns the price to the main program.

Finally, the chosen item, its price, computed tax and tip, and their sum is printed to a file called "Receipt.txt" in a nice format. I use the available, standard `prettyPrint` functions to format the output and print it nicely.

2 Functions

1.
 - `main`
 - Prototype: `int main()`
 - Called by: the Operating System
 - Calls: `displayChoices`, `selectFile`, `printSummary`
 - Parameters: none
 - Returns: An integer denoting success or failure to the operating system
 - Displays top level menu by calling `displayChoices`, handles top level selection and then calls `selectFile` with the name of the appropriate file ("appetizer.txt", "entree.txt", "dessert.txt"), and appropriate corresponding text string ("Appetizer", "Main Entree", "Dessert") to display.
2.
 - `displayChoices`
 - Prototype: `void displayChoices()`
 - Called by: `main`
 - Calls: none
 - Parameters: none
 - Returns: void
 - Displays top level menu of three choices (See Figure 1).
3.
 - `selectFile`
 - Prototype: `void selectFile(string level2Name, string filename)`
 - Called by: `main`
 - Calls: `exit` on error

- Parameters: `string level2Name`, `string filename`
 - Returns: An integer: `price`
 - This function displays the `level2Name`. Opens `filename` for reading, checks for errors in the open, and reads from and displays the three item choices and their prices. Once the user picks an item, the function returns the price of the item to `main`. If the user makes an invalid choice this returns `-1`. It also opens for writing and writes the name of the picked item to "Receipt.txt." If there is an error in opening this file, the function prints an error message and calls `exit(-1)` to exit the program.
- 4.
- `printSummary`
 - Called by: `main`
 - Calls: `prettyPrint` functions and `exit` on error
 - Parameters: `double price`, `double tip`, `double tax`
 - Returns: `void`
 - This function prettily prints the receipt or bill to "Receipt.txt." Note that this function opens "Receipt.txt" for **appending** and checks for errors. If there is an error, the function prints an error message and calls `exit(-1)` to exit the program. It then prints `price`, `tip`, `tax`, computes their sum, and prints this as the total amount to be paid.
- 5.
- `prettyPrint` functions
- The `prettyPrint` functions are documented at <http://www.cse.unr.edu/~cs135>