|  |  |
| --- | --- |
| /\* |  |
|  | Project 1 CIS 17A |
|  | Group name:S.E.C |
|  | Group members: Brandon B., Alberto C. |
|  | Purpose: Coffee shop customer ordering menu. |
|  | \*/ |
|  |  |
|  | #include <iostream> |
|  | #include <fstream> |
|  | #include <string> |
|  | #include <vector> |
|  | #include <iomanip> |
|  | #include <stdlib.h> |
|  | #include <cctype> |
|  | #include <limits> |
|  | using namespace std; |
|  |  |
|  | const int row = 2, col = 3; |
|  | void displayMenu(); |
|  | void displayFoods(); |
|  | void displayDrinks(); |
|  | void displayMenuOrder(const string[][col], const double[][col], int); |
|  | void displayReceipt(int); |
|  | // Global values for food and drink items & prices |
|  | // So they can be referenced throughout the program as needed. |
|  | string items[row][col] = {{"Cappuccino","Latte" ,"Espresso" }, {"Biscotti","English Muffin","Butter Croissant"}}; |
|  | double itemPrice[row][col] = {{3.65,2.95 ,3.94 }, {1.50, 1.65, 1.75}}; |
|  | // Empty vectors to hold items to be placed in the receipt |
|  | vector<string> storeItem; |
|  | vector<double> storePrice; |
|  | vector<int> storeNumber; |
|  | double totalAccumulator = 0; |
|  | static int itemCount = 0; |
|  | const double TAX = 0.085; |
|  | int numWanted; |
|  | int count; |
|  |  |
|  | ofstream outputfile; |
|  |  |
|  | int main() |
|  | { |
|  | char check; |
|  | int i = 0; //used for creating recipt as vector sizes for all 3 vectors |
|  | double cost; |
|  | string input; |
|  | int choice; |
|  | int userInput; |
|  | double totalPrice = 0.0; |
|  | //Program Start// |
|  | cout << fixed << setprecision(2) << showpoint; |
|  | cout << "----------------------------------------------------------" << endl; |
|  | cout << "||\t\t\t\tWelcome to the Velvet Cafe.\t\t\t\t||" << endl; |
|  | do |
|  | { |
|  | if (userInput == 5) |
|  | { |
|  | displayReceipt(i); |
|  | return 0; |
|  | } |
|  | displayMenu(); |
|  | cin >> userInput; |
|  | cout << endl; |
|  | while(userInput < 0 || userInput > 3 ||!cin) |
|  | { |
|  | cout << "Regrettably, that is not one of the options.\n"; |
|  | cin.clear(); |
|  | cin.ignore(numeric\_limits<streamsize>::max(), '\n'); |
|  | cout << "Enter an option between 1-3: "; |
|  | cin >> userInput; |
|  | } |
|  |  |
|  | switch(userInput) |
|  | { |
|  | //for food menu |
|  | case 1 :do |
|  | { |
|  | displayFoods(); |
|  | cin >> userInput; |
|  | while(!cin || userInput < 1 || userInput > 5) //to check for number input |
|  | { |
|  | cout << "That option does not exist. Try again "; |
|  | cin.clear(); |
|  | cin.ignore(numeric\_limits<streamsize>::max(), '\n'); |
|  | cin >> userInput; |
|  | } |
|  | if(userInput == 4) //to return to main menu |
|  | { |
|  | break; |
|  | } |
|  | else if(userInput == 5) //to end order |
|  | { |
|  | break; |
|  | } |
|  | choice = userInput - 1; |
|  | input = items[0][choice]; |
|  | storeItem.push\_back(input); |
|  | cout << "\nHow many will you be purchasing? "; |
|  | while(!(cin >> count) || cin.peek() != '\n') //check for number input |
|  | { |
|  | cout << "That is not possible. Try again "; |
|  | cin.clear(); |
|  | cin.ignore(numeric\_limits<streamsize>::max(), '\n'); |
|  | } |
|  | //store into vector |
|  | storeNumber.push\_back(count); |
|  | cost = itemPrice[0][choice] \* count; |
|  | storePrice.push\_back(cost); |
|  | cout << "\nWill you be ordering another item?" << endl; |
|  | cout <<"Enter Y or y to order another item. "; |
|  | cin >> check; |
|  | cout << endl; |
|  | i++; |
|  | }while (check == 'Y' || check == 'y'); // Will loop back on this menu is 'y' is chosen |
|  | break; |
|  | //for drink menu |
|  | case 2 :do |
|  | { |
|  | displayDrinks(); |
|  | cin >> userInput; |
|  | while(!cin|| userInput < 1 || userInput > 5) //to check for number input |
|  | { |
|  | cout << "That option does not exist. Try again "; |
|  | cin.clear(); |
|  | cin.ignore(numeric\_limits<streamsize>::max(), '\n'); |
|  | cin >> userInput; |
|  | } |
|  | if(userInput == 4) //to return to main menu |
|  | { |
|  | break; |
|  | } |
|  | else if(userInput == 5) //to end order |
|  | { |
|  | break; |
|  | } |
|  | choice = userInput - 1; |
|  | input = items[1][choice]; |
|  | storeItem.push\_back(input); |
|  | cout << "\nHow many will you be purchasing? "; |
|  | while(!(cin >> count) || cin.peek() != '\n') //to check for number input |
|  | { |
|  | cout << "That is not a valid input. Try again "; |
|  | cin.clear(); |
|  | cin.ignore(numeric\_limits<streamsize>::max(), '\n'); |
|  | } |
|  | //store into vector |
|  | storeNumber.push\_back(count); |
|  | cost = itemPrice[1][choice] \* count; |
|  | storePrice.push\_back(cost); |
|  | cout << "\nWill you be ordering another item?" << endl; |
|  | cout <<"Enter Y or y to order another item."; |
|  | cin >> check; |
|  | cout << endl; |
|  | i++; |
|  | }while (check == 'Y' || check == 'y'); // Will loop back on this menu is 'y' is chosen |
|  | break; |
|  | //for order to end/show receipt |
|  | case 3:displayReceipt(i); |
|  | return 0; |
|  | } |
|  | cout << endl; |
|  | }while(userInput != 0); //to have user continue to order |
|  | return 0; |
|  | } |
|  |  |
|  | void displayMenu() |
|  | { |
|  | cout << "----------------------------------------------------------" << endl; |
|  | cout << "Select a menu from the options we have provided." << endl; |
|  | cout << "1. Food" << endl; |
|  | cout << "2. Drinks" << endl; |
|  | cout << "3. Finish Ordering" << endl; |
|  | cout << "0. Exit" << endl; |
|  | cout << "Now, your choice? "; |
|  | } |
|  |  |
|  | void displayFoods() |
|  | { |
|  | displayMenuOrder(items, itemPrice, 1); |
|  | } |
|  |  |
|  | void displayDrinks() |
|  | { |
|  | displayMenuOrder(items, itemPrice, 0); |
|  | } |
|  |  |
|  | void displayMenuOrder(const string item[][col], const double price[][col], int row) |
|  | { |
|  | cout << setw(20) << left << "ITEM" << "\t\t\t\t\tPRICE" << endl; |
|  | cout << "---------------------------------------------" << endl; |
|  | for (int i = 0; i < 3; i++) |
|  | { |
|  | cout << (i + 1) << ". " << setw(20) << left << item[row][i] << "\t\t\t\t\t$" << price[row][i] << endl; |
|  | } |
|  | cout << "4. Return to Main Menu" << endl; |
|  | cout << "5. End Order" << endl; |
|  | cout << "Your choice? "; |
|  | } |
|  |  |
|  | void displayReceipt(int i) |
|  | { //Show user their order |
|  | double sum; |
|  |  |
|  | outputfile.open("Receipt.text"); |
|  | //output to screen |
|  | cout << endl << endl; |
|  | cout << "\t\t\t Velvet Cafe " << endl; |
|  | cout << "--------------------------------------" << endl; |
|  | for(int n = 0; n < i; n++) |
|  | { |
|  | cout << storeNumber[n] << " " << setw(16) << storeItem[n] << "\t\t\t\t $" << storePrice[n] << endl; |
|  | } |
|  |  |
|  | //output to file |
|  | outputfile << "\t\t\t Velvet Cafe " << endl; |
|  | outputfile << "-------------------------------" << endl; |
|  | for (int n = 0; n < i; n++) |
|  | { |
|  | outputfile <<left << storeNumber[n] << " " << setw(16) << storeItem[n] << "\t\t\t\t$" << setprecision(2) << fixed << storePrice[n] << endl; |
|  | } |
|  |  |
|  | for(int n = 0; n < i; n++) |
|  | { |
|  | sum += storePrice[n]; |
|  | } |
|  | //output to screen |
|  | cout << endl; |
|  | // Minor changes made to allow for higher subtotals to still be in line with the receipt |
|  | cout << "SUBTOTAL\t\t\t\t\t\t$" << sum << endl; |
|  | cout << "TAX\t\t\t\t\t\t\t\t 8.5%" << endl; |
|  |  |
|  | //output to file |
|  | outputfile << endl; |
|  |  |
|  | outputfile << "SUBTOTAL\t\t\t\t\t\t\t\t $" << sum << endl; |
|  | outputfile << "TAX\t\t\t\t\t\t\t\t\t\t\t\t 8.5%" << endl; |
|  |  |
|  | sum = (sum \* TAX) + sum; |
|  | //output to screen |
|  | // Changes made to allow for higher totals to still be in line with the receipt |
|  | cout << left << setprecision(2) << fixed << "TOTAL\t\t\t\t\t\t $" << setw(5) << sum << endl; |
|  |  |
|  | //output to file |
|  | outputfile << left << setprecision(2) << fixed << "TOTAL\t\t\t\t\t\t\t\t\t\t $" << setw(5) << sum << endl; |
|  | outputfile.close(); |
|  | } |