**­CSC 1101 – Problem Solving and Programming Laboratory**

**Lab 17 – Omar Faruk**

**25 points – Due November 16, 11pm**

You've been hired by *Best Bargains* to write a C++ console application that calculates sale information. Use a sentinel loop (sentinel value “n”) to get the option to process another sale. Before the loop, initialize a sale ID to zero. If option 'y' entered, use a validation loop to get one of the following product codes:

|  |  |  |
| --- | --- | --- |
| **Product code** | **Product name** | **Product category** |
| **b** | Bluetooth speakers | Audio |
| **d** | Dishwasher | Appliances |
| **f** | File cabinet | Furniture |
| **h** | Headphones | Audio |
| **m** | Microwave | Appliances |
| **o** | Office chair | Furniture |

Then convert the product code to its corresponding product name by calling function productString (described below). Convert the product code to its corresponding product category by calling function categoryString (described below). Then use a validation loop to get a wholesale price at least zero. Use a validation loop to get a retail price at least the wholesale price. Then calculate the following:

● Increment the sale ID by one.

● The profit forselling the product (retail – wholesale).

● The 6% Michigan sales tax on the appliance (retail \* 0.06).

● The total cost to the customer (retail + sales tax).

Use formatted output manipulators (setw, left/right) to print the following rows:

● Sale ID

● Product

● Category

● Wholesale price ($)

● Retail price ($)

● Profit ($)

● Sales tax ($)

● Total cost ($)

And two columns:

● A left-justified label.

● A right-justified value.

**string productString(char c)**

This value function takes the product code and returns the corresponding product name. For example, code 'f' should return "File cabinet". This function does no printing, it just returns a string to the function that called it.

**string categoryString(char c)**

This value function takes the product code and returns the corresponding category. For example, code 'f' should return "Furniture". This function does no printing, it just returns a string to the function that called it.

Define constants for the sales tax rate and column widths. You could also define a constant for your product code prompt since it is long and used twice. Format all real numbers to two decimal places. Continue to prompt the user for sales until they enter sentinel value “n”. The output should look like this:

Welcome to Best Bargains

------------------------

Get a sale (y or n)? y

Enter a product (

b-Bluetooth speakers

d-Dishwasher

f-File cabinet

h-Headphones

m-Microwave

o-Office chair): b

Enter the wholesale price (>= $0): -30

Error: the wholesale price must be at least $0.

Enter the wholesale price (>= $0): 150

Enter the retail price (>= $150.00): 100

Error: the retail price must be at least $150.00.

Enter the retail price (>= $150.00): 200

Sale: 1

Product: Bluetooth speakers

Category: Audio

Wholesale price ($): 150.00

Retail price ($): 200.00

Profit ($): 50.00

Sales tax (6%): 12.00

Total cost ($): 212.00

Get another sale (y or n)? y

Enter a product (

b-Bluetooth speakers

d-Dishwasher

f-File cabinet

h-Headphones

m-Microwave

o-Office chair): m

Enter the wholesale price (>= $0): 300

Enter the retail price (>= $300.00): 400

Sale: 2

Product: Microwave

Category: Appliances

Wholesale price ($): 300.00

Retail price ($): 400.00

Profit ($): 100.00

Sales tax (6%): 24.00

Total cost ($): 424.00

Get another sale (y or n)? n

End of Best Bargains

Run the program with at least three sales of different sets of inputs. Include some invalid inputs in the run. Do not use the sample input above for your final run pasted below.

*[your program code here]\**

//==========================================================

//

// Title: Best Bargains Sales

// Course: CSC 1101

// Lab Number: 17

// Author: Omar Faruk

// Date: 11/15/2020

// Description:

// Creating a sales application for Best Bargains to take sales

// and calculate price, profit, and cost using functions,

// swtch statements, and validation and sentinel loops.

//

//==========================================================

#include <cstdlib> // For several general-purpose functions

#include <fstream> // For file handling

#include <iomanip> // For formatted output

#include <iostream> // For cin, cout, and system

#include <string> // For string data type

using namespace std; // So "std::cout" may be abbreviated to "cout"

// Global

const int COLMFT1 = 45, COLMFT2 = 25;

const float state\_tax = .06;

// Function: product name

string productString(char c)

{

string product\_name;

switch (c)

{

case 'b':

product\_name = "Bluetooth speakers";

break;

case 'd':

product\_name = "Dishwasher";

break;

case 'f':

product\_name = "File cabinet";

break;

case 'h':

product\_name = "Headphones";

break;

case 'm':

product\_name = "Microwave";

break;

case 'o':

product\_name = "Office chair";

break;

default:

product\_name = "Unknown product code";

}

return product\_name;

}

// Function: product category

string categoryString(char c)

{

string product\_category;

switch (c)

{

case 'b':

product\_category = "Audio";

break;

case 'd':

product\_category = "Appliances";

break;

case 'f':

product\_category = "Furniture";

break;

case 'h':

product\_category = "Audio";

break;

case 'm':

product\_category = "Appliances";

break;

case 'o':

product\_category = "Furniture";

break;

default:

product\_category = "Unknown product category";

}

return product\_category;

}

int main()

{

// Declare variables

char process\_sale, product\_code;

string product\_name, product\_category;

float wholesale\_price, retail\_price, product\_profit, sales\_tax, total\_cost;

// Show application header

cout << "Welcome to Best Bargains!" << endl;

cout << "--------------------------" << endl << endl;

// Format real numbers to 2 decimal places

cout << setprecision(2) << fixed;

// Initialize sale ID

int sale\_id = 0;

// Sentinel loop, n for no sale & y for processing new sale

cout << "Get a sale (y or n)? ";

cin >> process\_sale;

while (process\_sale != 'n')

{

while (process\_sale == 'y')

{

// Sale ID

sale\_id++;

// Product code menu

cout << "\nEnter a product code:" << endl

<< "b - Bluetooth speakers" << endl

<< "d - Dishwasher" << endl

<< "f - File cabinet" << endl

<< "h - Headphones" << endl

<< "m - Microwave" << endl

<< "o - Office chair : ";

cin >> product\_code;

// Convert product code to corresponding name productString func

product\_name = productString(product\_code);

// Convert product code to corresponding category categoryString func

product\_category = categoryString(product\_code);

// Validation loop for wholesale price >= 0

cout << "\nEnter the wholesale price ($0 or more): ";

cin >> wholesale\_price;

while (wholesale\_price < 0)

{

cout << "Error: wholesale price must be atleast $0." << endl

<< "\nEnter the wholesale price ($0 or more): ";

cin >> wholesale\_price;

}

// Validation loop for retail price >= wholesale price

cout << "\nEnter the retail price ($" << wholesale\_price <<" or more): ";

cin >> retail\_price;

while (retail\_price < wholesale\_price)

{

cout << "Error: retail price must be atleast $" << wholesale\_price << "." << endl

<< "\nEnter the retail price ($" << wholesale\_price << " or more): ";

cin >> retail\_price;

}

// Calculations

product\_profit = retail\_price - wholesale\_price;

sales\_tax = retail\_price \* state\_tax;

total\_cost = retail\_price + sales\_tax;

// Screen output

cout << setw(COLMFT1) << left << "\nSale:"

<< setw(COLMFT2) << right << sale\_id << endl

<< setw(COLMFT1) << left << "Product:"

<< setw(COLMFT2) << right << product\_name << endl

<< setw(COLMFT1) << left << "Category:"

<< setw(COLMFT2) << right << product\_category << endl

<< setw(COLMFT1) << left << "Wholesale price ($):"

<< setw(COLMFT2) << right << wholesale\_price << endl

<< setw(COLMFT1) << left << "Retail price ($):"

<< setw(COLMFT2) << right << retail\_price << endl

<< setw(COLMFT1) << left << "Profit ($):"

<< setw(COLMFT2) << right << product\_profit << endl

<< setw(COLMFT1) << left << "Sales tax (6%):"

<< setw(COLMFT2) << right << sales\_tax << endl

<< setw(COLMFT1) << left << "Total cost ($):"

<< setw(COLMFT2) << right << total\_cost << endl;

//Prompt for another sale

cout << "\nGet another sale (y or n)? ";

cin >> process\_sale;

}

}

// Show application close

cout << "\nEnd of Best Bargains" << endl;

return 0;

}

*[your program output here]\*\**



