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

**Lab 12 – Omar Faruk**

**25 points – Due October 19, 11pm**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Paste your code and screenshots into the document.

**c)** Submit this document and your .cpp file(s) to the Canvas item where you downloaded this document. Do not submit a zip file but individually attach your files.

Remember that great app you wrote for *Yogurt Yummies* (Lab 10-2). They want you to enhance the C++ console application that calculates and displays the cost of a customer’s yogurt purchase. Now they need to process multiple sales. Do not change any logic for handling a single sale. Make the following enhancements:

● Add "v2" to the application output header and close.

● Declare and initialize overall totals including:

✓ Number of sales.

✓ Overall number of yogurts sold.

✓ Overall sale amount after discount.

✓ Overall tax paid.

✓ Overall sale total.

● Enclose the logic for a single sale with a sentinel loop that continues to process sales until the user enters 'n'.

● After calculating sale totals, update overall totals.

● When the user enters the sentinel value ('n'), print overall totals using formatted output manipulators (setw, left/right). Run the program with invalid and valid inputs, and at least three sales. The output should look like this:

Welcome to Yogurt Yummies, v2

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

Enter another yogurt purchase (y/n)? y

Sale 1

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

Enter the number of yogurts purchased (1-9): 11

Error: '11' is an invalid number of yogurts.

Enter the number of yogurts purchased (1-9): 2

Enter the percentage discount (0-20): 22

Error: '22.00' is an invalid percentage discount.

Enter the percentage discount (0-20): 4

Yogurts: 2

Yogurt cost ($): 3.50

Discount (%): 4.00

Subtotal ($): 7.00

Total after discount ($): 6.72

Tax ($): 0.40

Total ($): 7.12

Enter another yogurt purchase (y/n)? y

Sale 2

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

Enter the number of yogurts purchased (1-9): 5

Enter the percentage discount (0-20): 10

Yogurts: 5

Yogurt cost ($): 3.50

Discount (%): 10.00

Subtotal ($): 17.50

Total after discount ($): 15.75

Tax ($): 0.94

Total ($): 16.70

Enter another yogurt purchase (y/n)? y

Sale 3

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

Enter the number of yogurts purchased (1-9): 7

Enter the percentage discount (0-20): 20

Yogurts: 7

Yogurt cost ($): 3.50

Discount (%): 20.00

Subtotal ($): 24.50

Total after discount ($): 19.60

Tax ($): 1.18

Total ($): 20.78

Enter another yogurt purchase (y/n)? n

Overall totals

========================================

Sales: 3

Yogurts: 14

Total after discount ($): 42.07

Tax ($): 2.52

Total ($): 44.59

End of Yogurt Yummies, v2

Do not use this sample input for the final run pasted below.

*[your program code here]\**

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

//

// Title: Yogurt Shop Sales Calculator

// Course: CSC 1101

// Lab Number: Lab 12

// Author: Omar Faruk

// Date: 10/18/2020

// Description:

// Using validation loop for two inputs and calculations

// to create an yougurt sales calculator, then using a nested

// loop to take more than 1 sale and creating a final formatted

// total of all nested loop entries.

//

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

#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"

int main()

{

// Declare constants

const int COLMFT1 = 30;

const int COLMFT2 = 5;

const double YOGURT\_PRICE = 3.50;

const double SALES\_TAX = .06;

// Declare variables

int yogurt\_quantity;

double discount\_percentage;

double yogurt\_subtotal;

double yogurt\_discRate;

double yogurt\_discTotal;

double tax;

double yogurt\_total;

string option;

cout << fixed << setprecision(2);

// Show application header

cout << "Welcome to Yogurt Yummies! v2" << endl;

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

// Write to screen

cout << "Enter another yogurt purchase (y/n)? ";

cin >> option;

//Nested Loop

while (option != "n")

{

// Declaring variables

int sales = 1;

int numYogurt = 0;

double numDiscTotal = 0;

double numTotal = 0;

double numTax = 0;

while (option == "y")

{

cout << "\nSale " << sales++ << endl;

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

// Yogurt Quantity Validation Loop w/Input

cout << "\nEnter the number of yogurts purchased (1-9): ";

cin >> yogurt\_quantity;

while (yogurt\_quantity < 1 || yogurt\_quantity > 9)

{

cout << "Error: " << yogurt\_quantity << " is an invalid number of yogurts." << endl;

cout << "\nEnter the number of yogurts purchased (1-9): ";

cin >> yogurt\_quantity;

cout << endl;

}

// Discount Code Validation Loop w/Input

cout << "Enter the percentage discount (0-20) : ";

cin >> discount\_percentage;

while (discount\_percentage < 0 || discount\_percentage > 20)

{

cout << "Error: " << discount\_percentage << "% is an invalid percentage discount" << endl;

cout << "\nEnter the percentage discount (0-20) : ";

cin >> discount\_percentage;

cout << endl;

}

// Calculation

yogurt\_subtotal = (YOGURT\_PRICE \* yogurt\_quantity);

yogurt\_discRate = (discount\_percentage / 100);

yogurt\_discTotal = (yogurt\_subtotal - (yogurt\_subtotal

yogurt\_discRate));

tax = (yogurt\_discTotal \* SALES\_TAX);

yogurt\_total = (yogurt\_discTotal + tax);

// Formatted output for calculations

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

<< setw(COLMFT2) << right << yogurt\_quantity << endl;

cout << setw(COLMFT1) << left << "Yogurt cost ($): "

<< setw(COLMFT2) << right << YOGURT\_PRICE << endl;

cout << setw(COLMFT1) << left << "Discount (%): "

<< setw(COLMFT2) << right << discount\_percentage << endl;

cout << setw(COLMFT1) << left << "Subtotal ($)"

<< setw(COLMFT2) << right << yogurt\_subtotal << endl;

cout << setw(COLMFT1) << left << "Total after discount ($)"

<< setw(COLMFT2) << right << yogurt\_discTotal << endl;

cout << setw(COLMFT1) << left << "Tax ($)"

<< setw(COLMFT2) << right << tax << endl;

cout << setw(COLMFT1) << left << "Total ($)"

<< setw(COLMFT2) << right << yogurt\_total << endl;

//Adding input together for each nested loop

numYogurt = numYogurt + yogurt\_quantity;

numDiscTotal = numDiscTotal + yogurt\_discTotal;

numTax = numTax + tax;

numTotal = numTotal + yogurt\_total;

cout << "\nEnter another yogurt purchase (y/n)? ";

cin >> option;

}

//Overall Totals Output

cout << setw(COLMFT1)<< left << "\nOverall totals" << endl;

cout << "======================================" << endl;

cout << setw(COLMFT1) << left << "Sales:"

<< setw(COLMFT2) << right << sales - 1 << endl;

cout << setw(COLMFT1) << left << "Yogurts:"

<< setw(COLMFT2) << right << numYogurt << endl;

cout << setw(COLMFT1) << left << "Total after discount ($):"

<< setw(COLMFT2) << right << numDiscTotal << endl;

cout << setw(COLMFT1) << left << "Tax ($):"

<< setw(COLMFT2) << right << numTax << endl;

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

<< setw(COLMFT2) << right << numTotal << endl;

}

// Show application close

cout << "\nEnd of Yogurt Yummies v2" << endl;

}

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





