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

**Lab 10 – Omar Faruk**

**25 points – Due October 12, 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.

**1) [12 points]** You've been hired by *Avuncular Addresses* to write a C++ console application that analyzes and checks a postal address. Prompt for and get from the user an address. Use function **getline** so that the address can contain spaces. Loop through the address and count the following types of characters:

● Digits (0-9)

● Alphabetic (A-Z, a-z)

● Other

Use function **length** to control the loop. Use functions **isdigit** and **isalpha** to determine the character types. Use formatted output manipulators (setw, left/right) to print the following rows:

● Address

● String length

● Number of digits

● Number of alphas

● Number of other characters

And two columns:

● A left-justified label.

● A right-justified value.

Then test the number of digits and number of alphas. If digits is less than two or alphas is less than three, print an invalid address message. Otherwise, print a valid address message. Define constants for the minimum number of digits and alphas, and the column widths. The output should look like this for invalid and valid input:

Welcome to Avuncular Addresses

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

Enter an address: 1 West Liberty St

Address: 1 West Liberty St

Length: 16

Digits: 1

Alphas: 12

Other: 3

Address is invalid!

End of Avuncular Addresses

Welcome to Avuncular Addresses

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

Enter an address: 110 Main St

Address: 110 Main St

Length: 11

Digits: 3

Alphas: 6

Other: 2

Address is valid!

End of Avuncular Addresses

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

*[your program code here]\**

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

//

// Title: Address: Alpha, Digits, & Other Char Counter

// Course: CSC 1101

// Lab Number: Lab 10-01

// Author: Omar Faruk

// Date: 10/11/20

// Description:

// Count the alphas, digits, and others character

// using while loop and if else statement for validation print.

//

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

#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 = 25;

const int COLMFT2 = 20;

// Declare variables

string address;

char l;

char i;

int count\_alphas=0;

int count\_digits=0;

int count\_other=0;

// Show application header

cout << "Welcome to Avuncular Addresses!" << endl;

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

// Write to screen

// Getting user input

cout << setw(COLMFT1) << left << "Enter an address:";

getline(cin, address);

cout << endl;

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

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

// Loop to count alpha, digit, and other

l = address.length();

i = 0;

while (i < l)

{

if (isalpha(address.at(i)))

count\_alphas++;

else if (isdigit(address.at(i)))

count\_digits++;

else {

count\_other++;

}

i++ ;

}

// Printing out results

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

<< setw(COLMFT2) << right << address.length() << endl;

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

<< setw(COLMFT2) << right << count\_digits << endl;

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

<< setw(COLMFT2) << right << count\_alphas << endl;

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

<< setw(COLMFT2) << right << count\_other << endl;

// Check statement with if else for valid input

if (count\_digits < 2 || count\_alphas < 3)

cout << setw(COLMFT1) << left << "Address is Invalid!" << endl;

else {

cout << "Address is Valid!";

}

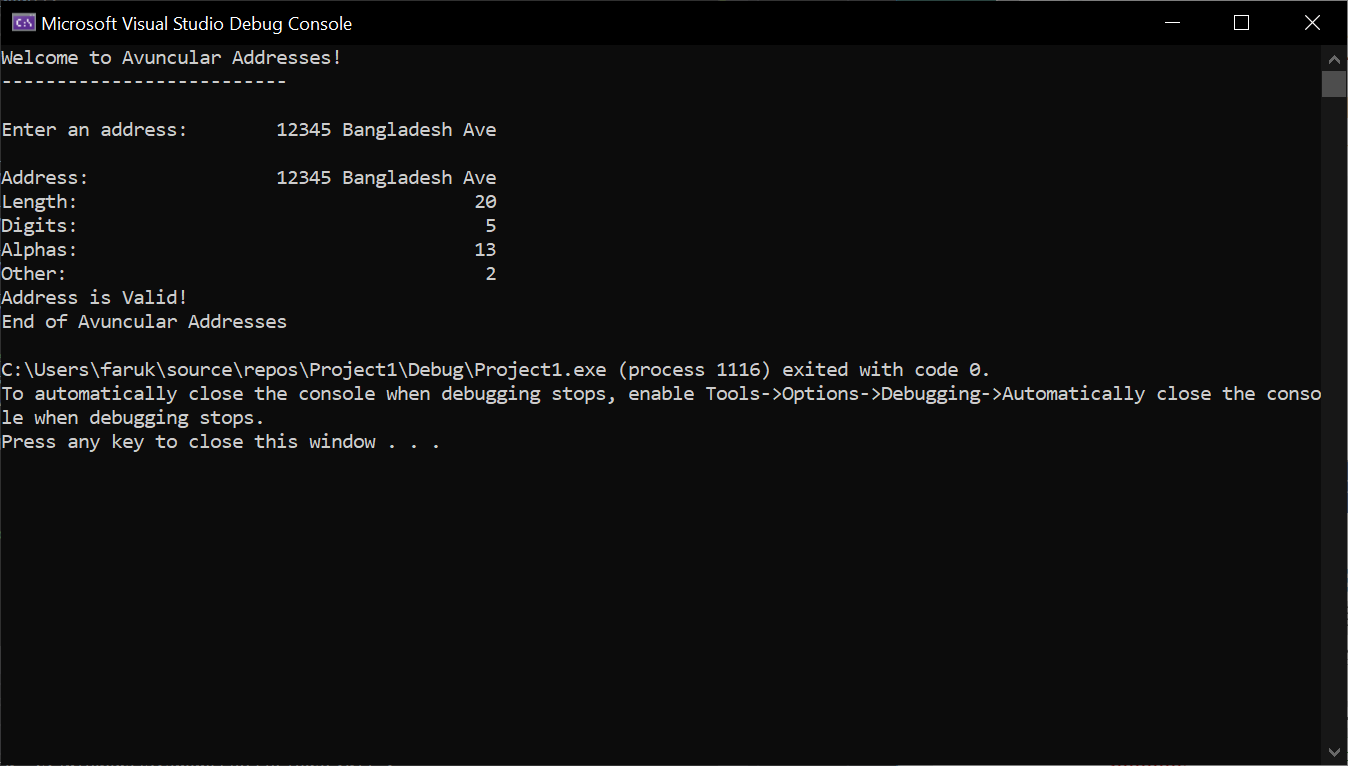
// Show application close

cout << "\nEnd of Avuncular Addresses" << endl;

}

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





**2) [13 points]** You've been hired by *Yogurt Yummies* to write a C++ console application that calculates and displays the cost of a customer’s yogurt purchase. Use a validation loop to prompt for and get from the user the number of yogurts purchased in the range 1-9. Then use a validation loop to prompt for and get from the user the coupon discount in the range 0-20%. Calculate the following:

● Subtotal using a cost of $3.50 per yogurt.

● Subtotal after discount

● Sale tax using rate of 6%.

● Total

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

● Yogurts purchased

● Yogurt cost ($)

● Discount (%)

● Subtotal ($)

● Subtotal after discount ($)

● Tax ($)

● Total ($)

And two columns:

● A left-justified label (including units)

● A right-justified value.

Define constants for the yogurt cost, sales tax rate, and column widths. Format all real numbers to two decimal places. Run the program with invalid and valid inputs. The output should look like this:

Welcome to Yogurt Yummies

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

Enter the number of yogurts purchased (1-9): 12

Error: '12' is an invalid number of yogurts.

Enter the number of yogurts purchased (1-9): 4

Enter the percentage discount (0-20): 30

Error: '30.00' is an invalid percentage discount.

Enter the percentage discount (0-20): 10

Yogurts: 4

Yogurt cost ($): 3.50

Discount (%): 10.00

Subtotal ($): 14.00

Total after discount ($): 12.60

Tax ($): 0.76

Total ($): 13.36

End of Yogurt Yummies

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 10-02

// Author: Omar Faruk

// Date: 10/11/20

// Description:

// Using validation loop for two inputs and calculations

// to create an yougurt sales calculator.

//

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

#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;

cout << fixed << setprecision(2);

// Show application header

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

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

// Write to screen

// Yogurt Quantity Validation Loop w/Input

cout << "Enter 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

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 ($)"

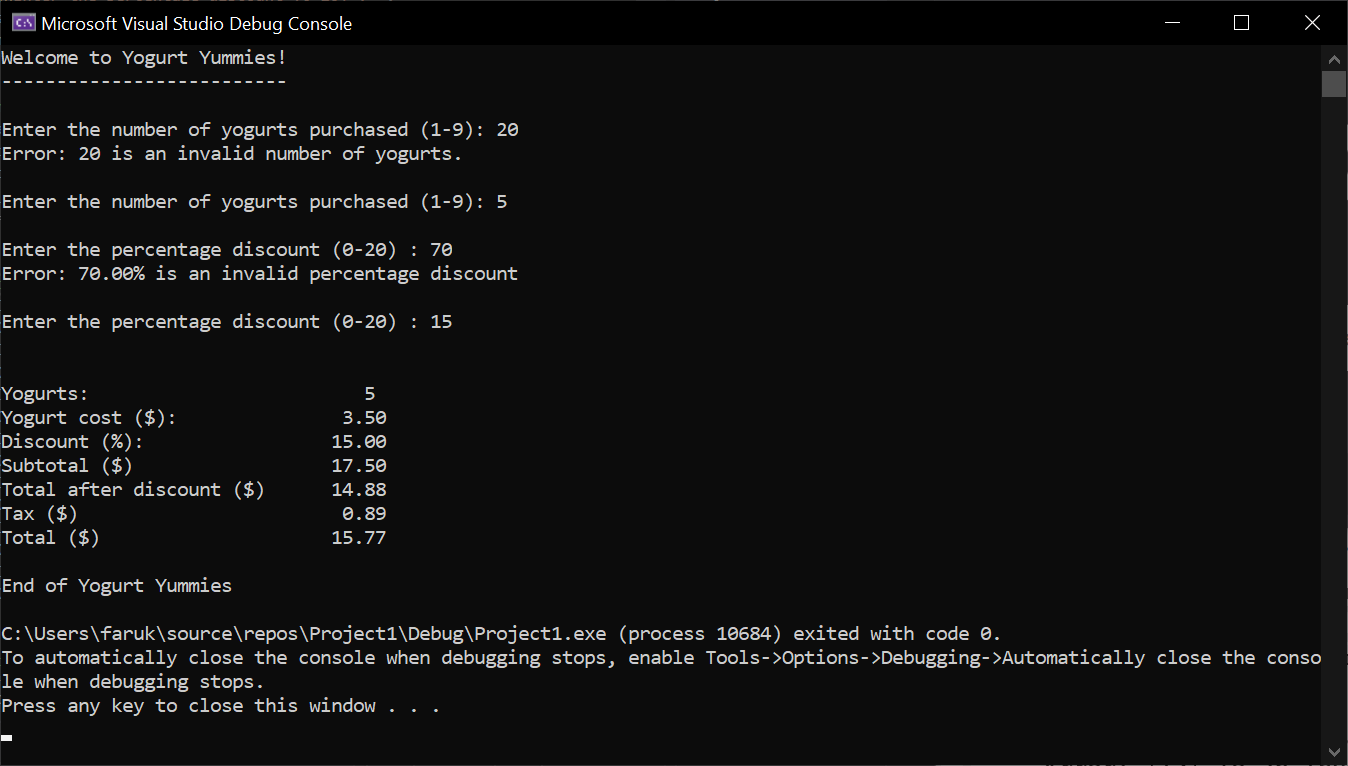
<< setw(COLMFT2) << right << yogurt\_total << endl;

// Show application close

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

}

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



**\* Copying-and-pasting C++ code to a Word document**

**macOS**

1) From within the C++ program, press **command-A** and press **command-C**.

2) From within the Word document, press **command-V**.

**Windows**

1) From within the C++ program, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

**\*\* Copying-and-pasting C++ console application output to a Word document**

**macOS**

1) From the C++ console, press **shift-command-4-space**.

2) From within the Word document, **command-V**.

**Windows**

1) From the C++ console, press **ALT-PrintScreen**.

2) From within the Word document, press **CTRL-V**.