**CS 2060 Programming with C - Fall 2017**

**Assignment #10**

Due Date: Nov 20, 2017 at 9:25am (MW class), Nov 21, 2017 at 9:25am (TR class)

Purpose: Learn more about files, arrays of strings, pointers to functions, and string functions.

Effort: Individual

Points: 100

Deliverables: Upload the .c source code file to Blackboard by due date.

Please include pseudocode in the comments at the beginning of your code.

**Please hand in a hardcopy version of your code at beginning of class.**

**Assignment Description**

For this assignment you will write a program that produces a letter based on the information in two files. The first file contains customer information and the second file contains a template for the letter. There are 3 possible letters that can be produced: confirmation, shipping and review letter. Each of these letters has a template file. To determine which letter to produce for a specific customer, a number is included as part of the customer data for each customer. The letters produced by your program should be written to new files.

Note that there is no console input for this program, it is completely driven by the information in the files. The number in the file (indication of what letter to produce) should be used as an index into an array of function pointers. These function pointers point to the different letter producing functions.

**Specifications**

1. Create a C project called **Assignment10 (please use this exact name)**
2. Follow "CS2060 Programming Assignments Policy"
3. Details:
   1. 4 files will be provided on Blackboard
      1. Customer data file
      2. Confirmation letter template file
      3. Shipping letter template file
      4. Review letter template file
   2. Customer data file format:
      1. For each customer there will be this **EXACT** information:
         1. A number which specifies which letter to produce for this customer

0 – confirmation letter

1 – shipping letter

2 – review letter

* + - 1. Seven strings that represent the customer data:

Product purchased

Customer name

Customer street address

Customer city

Customer state

Customer zip

Cost of order

* 1. Template letter file format:
     1. The template files contain the text to create a specific letter.
     2. Your program must read the template file, extract the lines of text, and fill in the customer information where specified.
        1. The places where you need to fill in customer information is represented with brackets. For example, in the confirmation letter template the first line is:

Dear [name]

**[name]** in template file needs to be replaced with the customer’s name from the customer data file. You will see other cases where you need to fill in the customer’s street address, city, state, etc.

* 1. The file names of the letters **produced by your code** must be:
     1. ConfirmationLetter.txt
     2. ShippingLetter.txt
     3. Review.txt
     4. Write files to a “Dev” directory on your C drive – **C:/Dev** (equivalent on MAC)

1. Write code that:
   1. Does **NOT** use global variables; you can and **SHOULD** use global constants.
   2. Uses array of strings to store the customer data – chapter 7
   3. Uses array of function pointers to call the different letter producing functions –chapter 7
   4. Uses functions from chapter 8 to process the file data.

**Tips**

* **DO NOT** **assume** that the number of customers is 3 based on the given customer data file. Your code may be tested on a shorter file.
* **DO assume** that only one of each type of letter will be created.
* Get the code working for one template file – the code for the others will be the same.
* If pointers to functions are confusing, get the code working first without the pointers. It will make more sense once everything is working to add in the array of pointers to the functions.
* Don’t go crazy with the functions from chapter 8. You should only need:
  + fgets
  + sscanf
  + strtok
  + strncmp
  + strcat
* Dealing with string data and the string functions can be challenging. Work on the code bit by bit to get the different string processing pieces working one at a time.
* Note: in the template files things like periods and $ are missing. I purposely did not include these so you would not have to deal with the extra processing for such items.

**Output**

One example showing the **customer data file**, the **confirmation letter template** and the **resulting confirmation letter**.

**CustomerData.txt** (given)

0

Programming With C

Larry Page

1600 Amphitheatre Parkway

Mountain View

California

94043

81.07

1

Data Structures and Algorithms in Java

Elon Musk

3500 Deer Creek Road

Palo Alto

California

94304

32.50

2

Computer Science Distilled

James Gosling

1234 Main Dr.

San Francisco

California

94122

35.70

**ConfirmationTemplate.txt** (given)

Hello **[name]**

Thank you for ordering **[product]**

Shipped To:

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

**[name]** These are the values your code will

**[streetaddress]** fill in with the information in the

**[city] [state] [zip]** customer data file.

Order Cost:

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

**[ordercost]**

You will receive another email when your order ships.

Visit us again. EducationalBooks.com

**ConfirmationLetter.txt** (produced by your program and written to a file)

Hello **Larry Page**

Thank you for ordering **Programming With C**

Shipped To:

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

**Larry Page**

**1600 Amphitheatre Parkway**  All bracket values [] have been

**Mountain View** replaced with the proper

**California** customer data.

**94043**

Order Cost:

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

**81.07**

You will receive another email when your order ships.

Visit us again. EducationalBooks.com