**CSE 1310 – Fall 20**

**Introduction to Programming**

**Lab 4**

Assigned: Thursday, November 26, 2020

#### Due: Sunday, December 13, 11:59pm.

####  \*\*ABSOLUTELY NO LATE LABS ACCEPTED FOR LAB4\*\*

*Extra Credit Early Submission: Wednesday, Dec 9, 11:59pm +20 points*

*Extra Credit Early Submission: Thursday, Dec 10, 11:59pm +15 points*

*Extra Credit Early Submission: Friday, Dec 11, 11:59pm +10 points*

*Extra Credit Early Submission: Saturday, Dec 12, 11:59pm +5 points*

The Lab assignment will be graded out of 100 points. There are multiple parts or tasks that make up each Lab.

Some tasks ask you to write code, and specify what name to use for the file in NetBeans. You need to use exactly the name that is given (do not change the case, or make any other modification). Remember, the name of the main class must match the filename.

There are further instructions at the bottom (after the questions) about how to save the file from NetBeans in order to be able to turn it in.

For every lab assignment you need to create an answers file. In this answers file you will put in answer any questions that are asked, you will show the output of code that you write and you will reference any code files that you create for a given question. See below for more details about what goes in the answers file.

Your answers document needs to be named with your initials and the last four digits of your ID number and then Lab#answers. So if my initials are JCMT and the last four digits of my ID are 1234, then the answers file for my Lab 4 would be **JCMT1234Lab3answers**.

* The ONLY acceptable file formats are Word document, OpenOffice document, and PDF.
* Put your last name, first name and UTA ID in the file on the first line.  *[-5 deduction if not\*\*]*
* Label the answers for each question with the number/letter of the question. Separate each answer from the next answer by at least two blank lines. *[-5 deduction if not\*\*]*
* Include EVERY question number/letter combination from the assignment in your answers document. If the question is a coding question telling you to save a file, for example some question numbered 17.b), then in your answers document you should have a line like the following for question 17.b):

17.b) Please see file Lab1Part3.c for this question.”

* Put all your question answers the answer document.
* If the lab question asks you to show the output of a doing some particular thing with the code, then you must also put a screenshot of the output in the answer document. For output that takes up more than one screen, make multiple pictures so that every screen is recorded. If you do not include the screenshots in your answer document, then the questions that should have had screenshots will be considered “Not answered” and will be awarded ZERO 0 points.

Each task below will instruct you where to put your answers. If the task says to “Save your program as file **XYZ1234Lab1Task1.c**” then this .c file should be turned in as part of the assignment along with the answers file.

Every lab assignment has a given due date. No late labs will be accepted. (Five minutes late is still late.) Lab assignments will be posted on Canvas. If you are unable to turn in your lab on time, you may request an extension by sending an e-mail to Dr Tiernan within 24 hours of the deadline for the lab. The e-mail must explain WHY you are not able to submit on time. If Dr. T grants an extension, she will reply to your message with the late penalties and new deadlines.

## **Instructions about C files, i.e. the .c documents:**

At the very top of the C file you must replace the first three lines of comments (the part that starts with “To change”) with something like “CSE1310 Fall 2020” . This is described as part of the Lab1 assignment.

Every document that is turned in MUST have your name and your student ID. This means that your .c files must have your name and ID number in the comments at the top of the file. You must put your name and ID just in the second comment block that NetBeans puts in your code. Below the label Author: and the creation date, put your name as specified in the instructions for this lab assignment and as shown above. This should be just above the include statements in the code. Ex:

/\*

 \* File: main.c

 \* Author: jcmtiernan

 \*

 \* Created on September 1, 2020, 12:57 PM

.\*

 \* Name: Dr. Carter Tiernan

 \* UTA ID: 1000000000

 \*/

*#include <stdio.h>*

In the empty comment block before main, make sure that you add comments to describe your own program. Here is an example

/\*

 \* Lab X Part Y – Testing arithmetic in C

 \*/

*int main…*

Every code file should have your initials and the last four digits of your ID number at the beginning of the file name. For example, if my initials are JCMT and the last four digits of my ID are 1234, then the C code file for my Lab 1 Task 1 would be **JCMT1234Lab1Task1.c** . In the questions below, the initials and digits are shown as **XYZ1234** . Every code file name should start with the initials and digits even if not explicitly listed. You will substitute your initials and digits for the **XYZ1234** part of the file name.

#### Summary of answer document and .c files prep

1. Create an answers file named with your initials and the last four digits of your ID number and then Lab#answers that is of file type .docx, .pdf, or .odt
2. Put lastname, firstname and UTA ID in the file then put every question number into the file.
3. Make sure to put some answer after every question number. The answer might be a written answer, a pasted pic or screenshot, or a message indicating that the answer is in an external file.
4. Make sure that when you create your C programs, your program includes your name and UTA ID in the Author: block. Also make sure your header comments describe your specific C program and you remove the default messages and put in your own messages.

###### There is info at the bottom about creating a folder for your assignment files to be turned in and submitting the assignment.

### Instructions that apply to all lab questions:

A. For later labs, if the instructions say to use code from previous lab/question then *everything* from previous question should still be in lab except what is explicitly changed in the current question.

B. \*\* Things not allowed in Dr. Tiernan’s 1310 class in C programs. {More details about these things will be posted in the file “Things Not Allowed”} If you use these elements in your lab assignment C code, you will get a 0 (zero) for the entire question where you used this element.

1) Cannot use *exit*

2) Cannot use *break* except between case statements in a switch structure

3) Cannot use *continue*

4) Cannot use an *intentional infinite loop* such as while (true)

5) Cannot use *goto*

**NOTE on compiler to use:**

I know that various folks – including myself – have been having issues with the NetBeans environment for C. If NetBeans isn't working here are some possible options. You are not limited to these choices but I've at least played with these a little.

OnlineGDB (onlinegdb.com/online\_c\_compiler )

JDoodle (jdoodle.com/c-online-compiler/ )

CodingGround (tutorialspoint.com/compile\_c\_online.php )

If a Surface Pro is all you have access to, then you can try the app Code Compiler Editor (<https://chrome.google.com/webstore/detail/code-compiler-editor/nagdmbckkknilkfndjaadheldefdkfdb?hl=en-US> ) to get some basic coding functionality for C programming.

You may have also used Eclipse, Code∷Blocks, etc. which are fine to use.

CAVEAT: Do not use any specialized commands that only work in that specific compiler.

**Grading Rubric for Lab 4:**

**Part 1: 100 points**

**Lab 4 parts to complete:**

**Part 1a:** You have been given some code Lab4Part1.c.

**1.a)** Create a new C application project called **XYZ1234Lab4Part1a**. Copy the code from Lab4Part1 into your new application. Make the modifications as described in the code and the questions below for Part 1a.

Modifications to make:

Meaningful names – Look for the comments related to meaningful names.

Replace generic / meaningless variable and function names with meaningful names

A "meaningful name" is a name that tells someone else what that data represents

Hard coded numbers – Look for the comments related to hard-coded numbers

A hard-coded number is a fixed numeric value that would have to be replaced with a different number in order to use the program for different data. Fixed values make your code less easy to reuse. At the top of the program you will see a bunch of constants defined. Use those constants to replace fixed values. Remember, if the value is a *variable* then it is not fixed. For example, using [5] is hard coded but using [x] is not because x is a variable.

In declarations - replace fixed numeric values with constant names defined below for sizes

In accessing array elements - replace fixed numeric values with constant names defined below for columns

Formatting and indentation – Indentation, placement of curly braces, and spacing primarily

Use the notes in the code to modify the code so that formatting and indentation is consistent throughout whole program

Use consistent spacing inside statements including assignments, arithmetic, control structures, and function calls

Insure that code is indented consistently throughout. Code within { } should be indented in one tab space from the {.

Opening { should either be consistently

a) on their own line, not at the end of the previous line OR

b) at the end of the previous line

Each level of nested { should then indent the code inside

Comments - Add comments as noted to explain various parts of the code

In some places, questions are given to help you write comments.

Some of these questions are also below in this document

Make sure that all output data is labeled so that a user would know what is represented by each set of data shown in the output. Make sure that your program runs completely without errors and the correct output is produced. Once your program is modified as required and working correctly, save your working program as **XYZ1234Lab4Part1a.c** [Don’t forget to put a note in your answer file for this question reminding the grader to look for the .c file {30 points}

*Rubric:*

*All meaningless/generic names {1} modified to more meaningful names {2} throughout program{1} {total 4 pts}*

*Correctly replace all hard-coded numbers with appropriate constants.{3} Choice of constants should be based on constant name {2} and program use {2} {total 7 pt}*

*Correctly indent as directed {3 pts}*

*Correctly and consistently use curly braces { } throughout code {3 pts}*

*Consistently use spacing throughout code {2 pts}*

*Add all comments as noted in the code {11 pts}*

**1.b)** Create a new C application project called **XYZ1234Lab4Part1b**. Copy the code from XYZ1234Lab4Part1a into your new application. Find the total cost of all the gifts and print it. Make the minimum modifications needed to the code to accomplish this.

Make sure that all output data is labeled so that a user would know what is represented by each set of data shown in the output. Make sure that your program runs completely without errors and the correct output is produced. Once your program is modified as required and working correctly, save your working program as **XYZ1234Lab4Part1b.c** [Don’t forget to put a note in your answer file for this question reminding the grader to look for the .c file ] {5 points}

*Rubric:*

*Correctly modify code to find the sum of the totals. {3 pts}*

*Print the correct total {2 pts}*

**1.b2)** Take a screenshot of the complete output of your program **XYZ1234Lab4Part1b.c** Make sure that the output information is understandable to the user. Paste the screenshot into your answers document as the answer to this question. {3 pts}

**1.c)** Create a second .txt data file called ***GiftList2.txt***that could be used as input for this program. You must use the same format as the given input file but have entirely new data. The data must be actual data that you found in an ad or a store or on the web. You must include at least 12 items in your data file. No more than 3 items may have free shipping. Tax must be either copied exactly from checkout OR may be calculated as 0.0825 times the price of the item. You may use the categories that are in the current data file and you may create additional categories as appropriate. Example categories might be "Electronics", "Antiques", "Food", etc. Have at least two items from each category you use and have at least 4 categories in your file. Save this data file ***GiftList2.txt*** as the file for this question. [Don’t forget to put a note in your answer file for this question reminding the grader to look for this text file ] {5 points}

**1.c2)** Run your **XYZ1234Lab4Part1b.c** program using this new file of data ***GiftList2.txt***. Take a screenshot of the complete program output. Make sure that the output information is understandable to the user. Paste the screenshot into your answers document as the answer to this question. {3 pts}

**1.d)** Create a new C application project called **XYZ1234Lab4Part1d**. Copy the code from XYZ1234Lab4Part1b into your new application. Write a bubble sort function to sort the arrays by price from lowest to highest. Keep both arrays sorted together. In the main function, print the arrays after running the bubble sort.

Make sure that your program runs completely without errors and the correct output is produced. Once your program is modified as required and working correctly, save your working program as **XYZ1234Lab4Part1d.c** [Don’t forget to put a note in your answer file for this question reminding the grader to look for the .c file ] {6 points}

*Rubric:*

*Correctly sort the arrays together. {3 pts}*

*Correctly sort based on price from lowest to highest {3 pts}*

**1.d2)** Take a screenshot of the complete output of your program **XYZ1234Lab4Part1d.c** Make sure that the output information is understandable to the user. Paste the screenshot into your answers document as the answer to this question. {3 pts}

**1.e)** Create a new C application project called **XYZ1234Lab4Part1e**. Copy the code from XYZ1234Lab4Part1d into your new application. Write a bubble sort function to sort the arrays by category alphabetically. If two items are in the same category, then sort within the category by item name. Keep both arrays sorted together. In the main function, print the arrays after running this bubble sort.

Make sure that your program runs completely without errors and the correct output is produced. Once your program is modified as required and working correctly, save your working program as **XYZ1234Lab4Part1e.c** [Don’t forget to put a note in your answer file for this question reminding the grader to look for the .c file ] {9 points}

*Rubric:*

*Correctly sort the arrays together. {3 pts}*

*Correctly sort based on category {3 pts}*

*Correctly sort within category based on item name {3 pts}*

**1.e2)** Take a screenshot of the complete output of your program **XYZ1234Lab4Part1e.c** Make sure that the output information is understandable to the user. Paste the screenshot into your answers document as the answer to this question. {3 pts}

**1.f)** Create a new C application project called **XYZ1234Lab4Part1f**. Copy the code from XYZ1234Lab4Part1e into your new application. Write a function to count the number of unique categories and to save the category names into an array. The category array that is created should end up with NO duplicate values. The function should take in the appropriate data and an empty array to hold the category names. The function should put the category names in the array and it should return a count of the number of unique categories that were found. In main, after calling this function, print the number of unique categories and print the values in the category array.

Make sure that your program runs completely without errors and the correct output is produced. Once your program is modified as required and working correctly, save your working program as **XYZ1234Lab4Part1f.c** [Don’t forget to put a note in your answer file for this question reminding the grader to look for the .c file ] {13 points}

*Rubric:*

*Correctly create function with required inputs and return type {2 pts}*

*Correctly find all unique categories and save in array {3 pts}*

*Correctly save each category only once, i.e. no duplicates in the array {2 pts}*

*Correctly count the categories {2 pts}*

*Correctly print count and category values in main {4 pts}*

**1.f2)** Take a screenshot of the complete output of your program **XYZ1234Lab4Part1f.c** Make sure that the output information is understandable to the user. Paste the screenshot into your answers document as the answer to this question. {3 pts}

**1.g)** Create a new C application project called **XYZ1234Lab4Part1g** Copy the code from XYZ1234Lab4Part1f into your new application. For this part you can either modify one of your existing functions or write a new function that will count the number of items in each category and store this information in a new *categoryCount* array. The *categoryCount* array will be declared in main and should store the numbers in the same order as the categories are listed in the category array that was created in part f. The new or modified function should take in the appropriate data and the empty *categoryCount* array to hold the category count. The function should put the category counts in the array. In main, after calling this function, print the name of each category and the number of items in that category.

Make sure that your program runs completely without errors and the correct output is produced. Once your program is modified as required and working correctly, save your working program as **XYZ1234Lab4Part1g.c** [Don’t forget to put a note in your answer file for this question reminding the grader to look for the .c file ] {10 points}

*Rubric:*

*Correctly create or modify function with required inputs and return type {3 pts}*

*Correctly count all occurences of each unique category and save in array {3 pts}*

*Correctly print category names and counts in main {4 pts}*

**1.g2)** Take a screenshot of the complete output of your program **XYZ1234Lab4Part1g.c** Make sure that the output information is understandable to the user. Paste the screenshot into your answers document as the answer to this question. {3 pts}

Miscellaneous: If you have questions, e-mail Dr. T and/or the TAs through Canvas.

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

**Suggestions**

1. Pay close attention to all requirements on this page, including file names and submission format. Even in cases where the program works correctly, points will be taken off for not following the instructions given on this page (such as wrong file names, wrong compression format for the submitted code, and so on). The reason is that non-compliance with the instructions makes the grading process significantly (and unnecessarily) more time consuming. Contact the instructor or TA if you have any questions.
2. For each new question that requires you to write code, make a new C project with the exact name that you need to use. Double check that there are no errors before submission. In case of an error in this situation, 0 points will be awarded.
3. Write your code in small steps. Do NOT try to write 25 lines of code and then start trying to run it. Instead, write a small chunk of code - 5 to 10 lines - then save and run those 10 lines to make sure they work and then continue. If the 5 to 10 lines have any errors, start on the errors closest to the top of the file and fix those first. Once you have made changes to fix ONE error, then run the code again to see if that really did fix the error. Don’t go forward until you have fixed to topmost error. If you can’t figure out how to fix that one – try to comment out the line with the error and then fix the remaining errors in your chunk. Don’t go further until you have fixed what you have so far.
4. If you are working on a project and want to start completely over (which is not always the best idea), copy the “old code” that you do not plan to reuse and save it into a text file in a directory with your other materials for the class. You may discover that you do want to reuse that code and if you have saved it, then you won’t have to rewrite all of it.

### **Instructions about preparing your lab to turn in (same as Lab1 with some extra suggestions):**

##### *How to prepare your assignment to turn in (or to "submit")*

Create a folder for your lab in your CSE 1310 class directory (outside of the Netbeans projects folder) and name the lab folder with your initials, your last 4 digits, and LabX. [If you don’t have your files organized into folders, I STRONGLY suggest that you start doing this. This is a way to sort the material you keep on your computer and be able to find it later. Check the internet for suggestions on how to do this type of organization.)

As an example, if your name is Happy Camper 1010101010, then your Lab 1 folder name is HXC1010Lab1. Save your HXC1010Lab1answers file in this folder. Next save all the .c files for the assignment in the folder. Below is how to do that. Once you have put all the needed .c files in your folder, zip the folder to compress it before submitting it (see info further down with submission details).

##### *Saving .c files out of NetBeans*

Go to the *main.c* file (the program) in NetBeans that you want to save. Make sure the program is open in the main window and is the program you are looking at. You should see the window tab highlighted for the program you want. For example, if you are saving the .c file for Lab1Part5 you will see the window tab for that file will say lab1part5/Lab1Part5. Make sure your cursor clicks in the editing window of that file.

From that window, go to the File menu and select Save As. This will bring up the directory where the file is currently stored (lab1part5 in NetBeansProjects). Navigate from the current directory to your new folder and save the file in the new folder. This will make a copy of the program in your new folder. You may notice now in NetBeans that the tab on the window of the program has changed. The tab might now say HXC1010Lab1/Lab1Part5. This shows that you have made a copy and stored it in the new directory. You should close this window in NetBeans.

NOTE: If you have closed a project in NetBeans and want to go back to it to edit it some more, you can go to the project by going to the Projects (or Files) tab either in the pane or on the left margin of the window. Then select the folder of the project you want. Open the folder and select Source Files (might say *src*), then select the name of the main.c or other source file you want, then click on the that file. This will open the program again in the editing window for you to keep working on.

TIP: If you open the .c file that you saved in your folder (HXC1010Lab1/Lab1Part5) you can edit it in NetBeans but NetBeans CANNOT run it. This is because NetBeans expects all the files to be in its own folder. What will happen is that you will make changes but when NetBeans runs, it will run the copy that is stored in NetBeans not the copy from your folder so none of your edits will be executed. This is why you should always go back to the files in NetBeans to keep working.

C FILE NOTE: Only the .c files and the answers file should be included in your zipped folder. Do NOT include the entire NetBeans project. You will be penalized if you save the project instead of just the .c file.

Because C names all of the main files as main.c, you will need to save each unique main.c into its own folder and then save these folders into your XYZ1234Lab1 folder for submission. Thus, if you are turning in Lab 1 with three parts, then you should have a XYZ1234Lab1 folder and then inside that folder you should have folders for Parts 1, 2, and 3, i.e. a XYZ1234Lab1Part1 folder, a XYZ1234Lab1Part2 folder, and a XYZ1234Lab1Part3 folder.

**How to submit**

The assignment should be submitted via Canvas. Submit a ZIPPED directory/folder called XYZ1234Lab1.zip. The file must be ZIPped (not RAR). No other forms of compression accepted. (Contact the instructor or TA if you do not know how to produce .zip files). The zipped directory should contain your answers document and all the C code files (task1.c etc).

To create a zipped directory called XYZ1234Lab1.zip, follow these steps:

1. Create a folder called XYZ1234Lab1.

2. Copy to that folder all your solutions (your answers file, all your C files, and any .txt files that were required for input or output).

3. Zip that folder. On some Windows systems, you can zip a folder by right-clicking on the folder, and then selecting Send to->Compressed (zipped) folder. On Mac, go to the parent directory of your Lab# directory in a Finder window, click on your XYZ1234Lab# folder, then select Compress from the list of file actions (under the gear icon).

4. Submit your zipped folder through your Canvas account at the lab assignment. You click on the name of the assignment and then it goes to the screen where you can upload your zipped file. (Assignments are only accepted through Canvas.)

5. Check that what you uploaded is exactly what you want by going into Canvas and then downloading the assignment you just uploaded, unzipping the folder, and checking the contents to make sure that everything you want to be there is present.

**Submission checklist**

• Did you answer all of the questions in the lab assignment and did you create all the required .c program files and text files, if any?

* Do all your .c program files run without errors in Netbeans?
* Do all your .c program files run and give correct answers in Netbeans?

• Did you create the answers file with your name and UTA ID in the file, and your answers to non-programming tasks?

* Did you include all the needed screenshots for output making sure that each screenshot was large enough to be read but small enough to keep the answers document from becoming huge?

• Did you put all of the files you created into one folder called XYZ1234Lab# ?

• Did you zip that folder into a file called XYZ1234Lab#.zip?

• Did you upload the zipped file to Canvas before the due date and time?

* Did you check what you uploaded to Canvas to make sure it has the desired material in it?