ITSC 2181 – Introduction to Computer Systems
Spring 2024
Module 03 – Unit 3: Lab

Objectives

- Practice how to write small C programs, including identifying and correcting `C` compiler error messages and warnings.
- Practice the use of `struct`s to organize related data.
- Practice the use of **arrays** in C to store and process data.
- Practice the use of **loops** in C to implement flow of control.
- Practice the use **strings** in C to store and process data..
- Practice the use of **functions** in C to organize code.

General Instructions

- Please do not write your email or user ID anywhere in the program. To identify your code, you may use your UNC Charlotte 800#

- In this lab, you will write a few short C programs. Each program needs to be in its own source code file, i.e., a file with the `.c` extension.

- **You need to test your code thoroughly before submitting it**.

- **To earn any credit, a program has to compile.** *You may comment-out lines that have errors to obtain partial credit for work done.*

- Programs need to compile cleanly to receive full credit, i.e., they do not produce any errors or warnings.

- It is essential that you **do your own work.**

    - **Do not use any external resources** (Internet, AI, friends, etc.). All cases of cheating will be taken very seriously.

    - **If you need help, please ask the instructor, TA/IA or CCI Tutoring center.**

Program 1: Customer Database (100 points)

Write a program, named `customer_db.c`, that reads and processes customer records (data about customers). Each **record** must be stored using a `struct` named `customer` that contains the following fields:

- **First Name**
  - All alphabetic.
  - Max of 20 characters long.
- **Middle Name**
  - All alphabetic.
  - Max of 20 characters long.
- **Last Name**
  - All alphabetic.
  - Max of 20 characters long.
- **Phone**
  - All alphabetic.
  - Max of 16 characters long.
  - Any format is acceptable.
- **Balance**
  - A floating point number.

The database consists of a collection of customer records, stored in an array of `struct`s. You may assume that this program is for a very small business, so **no more than 20 customer records need to be stored**. However, the exact number is not known beforehand.

- Users will type the data for each customer, one record at a time and following prompts.

- Each field is entered separately and there is a **new-line character at the end**, i.e., only one value per line.

- The program must always check the value entered for the *First Name* field and respond to one of two commands:
  :X Exits the program (**not** case sensitive)
  :S Shows the contents of the database (**not** case sensitive)

  Hints:
  - *Use an `if-else` conditional to identify the command. Also, note the semicolon.*
  - *Remember that strings in C are stored as arrays of individual characters. You can access each character by its array subscript.*

Sample Data

```
Kyle N Jones
1-904-123-1234
2500.25

Eileen Maria Garces
1-905-123-1235
650.95
```

- Assume that users will always enter correct data and that a value will be provided for every field.
- Note that individual fields do **not** contain spaces. *We will learn how to handle this later.*

Implementation

- The program must include the following user-defined functions:
```
void show_customer(struct customer cust);
void show_database(struct customer cust_db[], int size);
```

- The **`show_customer`** function displays the contents of one customer record on the console.

- The **`show_customer_db`** function displays the contents of the entire customer database on the console.

- The number of customers is not known in advance and your program cannot ask the user how many, but you may assume that there will not be more than twenty (20) customers in the input.

- The program must continue processing customer records until the user enters "**:X**" to quit the program.

Output

The program's output needs to be formatted as in the sample runs shown below. User input is in **bold** style.

Sample #1

```
Please enter the next customer record.
```

```
   :X    Exits the program
   :S    Shows the contents of the database

First Name: :X
Good bye!
```

Sample #2

```
Please enter the next customer record.
   :X    Exits the program
   :S    Shows the contents of the database

First Name: Kyle
Middle Name: N
Last Name: Jones
Phone Number: 1-904-123-1234
Balance Due: 2500.25

Please enter the next customer record.
   :X    Exits the program
   :S    Shows the contents of the database

First Name: :S

Customer List:
--------------
Customer: Kyle N Jones
Phone Number: 1-904-123-1234, Balance: $2500.25

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

Please enter the next customer record.
   :X    Exits the program
   :S    Shows the contents of the database

First Name: :x
Good bye!
```

Sample #3

```
Please enter the next customer record.
   :X    Exits the program
   :S    Shows the contents of the database

First Name: Kyle
Middle Name: N
Last Name: Jones
Phone Number: 1-904-123-1234
Balance Due: 2500.25
```

```
Please enter the next customer record.
   :X    Exits the program
   :S    Shows the contents of the database

First Name: Eileen
Middle Name: Maria
Last Name: Garces
Phone Number: 1-905-123-1235
Balance Due: 650.95

Please enter the next customer record.
   :X    Exits the program
   :S    Shows the contents of the database

First Name: :s

Customer List:
--------------
Customer: Kyle N Jones
Phone Number: 1-904-123-1234, Balance: $2500.25

--------------
Customer: Eileen Maria Garces
Phone Number: 1-905-123-1235, Balance: $650.95

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

Please enter the next customer record.
   :X    Exits the program
   :S    Shows the contents of the database

First Name: :X
Good bye!
```

Optional Challenge

Implement a command and corresponding user-defined function that calculates the outstanding balance owed by all the clients to the company.

Submission Instructions

1. Create a folder (directory) on your computer.

2. Name the folder **ITSC_2181_M03_U3_Lab_*student-id*** Replace *student-id* with your UNCC student ID (800#), e.g. *8001231234*

3. Download and copy your program (source code) files into this folder. You should keep the files for every lab in a separate folder (directory)

from your other course materials.

4. Upload your C files (individually) to *Canvas* and submit the assignment.

Grading Rubric

- This lab is worth a total of **100 points**.

- **Do your own work. Do not use any external resources** (Internet, friends, etc.). All cases of cheating will be taken very seriously. **If you need help, please ask the instructor, TA/IA or CCI Tutoring center.**

- Programs need to compile to earn any credit. *You may comment-out lines that have errors to obtain partial credit for work done.*

- Your work will be graded on three (3) major components: Logic and flow of program, output, and formatting/organization. Refer to the following table for details.

Logic and Flow of Program - 60%	
Fully Correct and code compiles without errors.	Full Credit
Minor Errors and code compiles without errors OR some required functionality is missing.	75% Credit
Major Errors and/or code compiles with warnings OR significant portions of the required functionality are missing.	50% Credit
Completely incorrect/missing/does not compile.	No Credit
Output - 30%	
Fully Correct and matches formatting and layout of sample output provided.	Full Credit
Minor Errors.	75% Credit
Major Errors.	50% Credit
Completely incorrect output.	No credit
Formatting/Organization of Code - 10%	
Code is clear, easy to read and formatter according to the guidelines. Whitespace has been used appropriately, including indentation and blank lines. Comments are used when needed.	Full Credit
Needs minor improvement.	75% Credit
Needs major improvement.	50% Credit
No formatting/organization at all.	No Credit

Additional Deductions

- Code that produces warnings: -10% per program
- Incorrectly named files: -2 points per file
- *Students who cheat on <u>any</u> course assignment, lab, test or other activity will have their course grade reduced by one letter grade, regardless of the activity's point value, if it is their first offense at UNC Charlotte.*