COP2220 Fall 2016 Project Phase I 11/13/2016

**This Phase will be the subject of constant updates. You are required to work on it alone. Specs will be clarified any time you seek clarification but help with your code will be provided only if you attempt what you need help with.**

**You are strongly advised to attend all sessions as all specs will be discussed in class.**

**Due Wednesday 11/23rd by 5:30pm via Blackboard (start of our regular meeting)**

**DO NOT submit your project if it is not working as specified. Instead keep it and work on it more, with a penalty of 8 points each session past 5:30pm.**

**Phase II will be posted as soon as all required tools are covered (partial specs are provided here)**

**All Phases I, II, are due the final day of classes (at 5:30am dead or alive)(please note that the final day of classes is the last session before our scheduled Final Exam session)**

**The whole project (all phases) requires your individual effort. This will be confirmed through multiple meeting with me when we will discuss your solution. You grade may drop, stay the same, increase, or be nullified.**

Code a complete C program solution to the following problem:

A tools rental company wants you to design an application to keep track of their day-to-day rental process to a set of customers.

This application requires the use of array of structure and must be handled by pointers (same tools and techniques used in class are required):

1-Code a structure Tool that has the following attributes with their related constraints:

**-Tool ID: exactly 5 digits**

**-Tool Type: letter and space only**

**-Full Day Rental: real number**

**-Half Day Rental: real number**

**-Rented //this will be either true if rented or false if available for rental**

Code a related function that input data for a given tool (will be used to add more tools to the database later)

Code a function that displays data for a given tool.

Code all your validation functions (to validate all attributes and based on their given constraints (type, length)

2-Code a structure Customer that has the following attributes with their related constraints:

**-Customer ID: 9 digits**

**-Customer First name: all letters.**

**-Customer Last name: all letters.**

**-Customer phone number: area code plus actual phone number (10 digits) (String)**

**-Customer email: String with the correct format (i.e.** [**abcd@gmail.com**](mailto:abcd@gmail.com)**) assume .com is the only valid extension.**

Code a related function that will be called to create a customer.

Code a related function that displays a given customer.

Code all your validation functions (to validate all attributes and based on their given constraints (type, length) (try recycling those validation functions used with other objects)

3-Code a structure Transaction that has the following attributes with their related constraints:

-Customer ID: same as in class Customer

-Tool ID: same as in class Tool

-Rental period: code as ‘F’ for full day or ‘H’ for half day. (generated)

-Rental fees: real number (calculated)

Code all your validation functions (to validate all attributes and based on their given constraints (type, length) (try recycling those validation functions used with other objects)

4-Code a main menu with the following options: (in base class)

1-Customer Menu

2-Tool Menu

3-Transaction Menu

4-Quit

5-Code a customer sub menu with the following options: (in base class)

1-Create a customer

2-Search a customer.

3-Go back to main menu

6-Code a tool sub menu with the following options: (in base class)

1-Add a tool

2-Search a tool

3-Goback to main menu

7- Code a transaction sub menu with the following options: (in base class)

1-Start a transaction

2-Search for a transaction

3-Display all transactions

4-Go back to main menu

8-Code a function that searches for a customer

9-Code a function that searches for a tool

10- Code a function that searches for a transaction by Customer ID (may get more than one hit)

11-Code a function that searches for a transaction by tool ID (one hit)

12-Code a function that searches for a transaction using both Customer ID and Tool ID

Can you make these searches more efficient…can you recycle some of it ,…combine…etc?

13-Code a function that generates the report for all transactions.

What to do?

-Declare an array for customers, an array for tools, and an array for transactions (size them accordingly)

-in main() call a function that will populate the array of Tools with hard coded attributes using the following data: (parallel dependencies!)

**Tool ID : CS012 , CC007, AC004, CM002, FS008**

**tool type : chain saw, concrete cutter, air compressor, cement mixer, floor sander**

**full day rental rate : 56.00 , 125.00, 39.00, 45.00, 75.00**

**half day rental rate : 45.00 , 95.00, 29.00, 39.00, 55.00**

**Rented : false, false, false, false, false**

**Important! When adding new tools, try using the same approach used for the ID …first two letters define the type of tool and the last 3 digits are for the actual tool number (example: CS012 refers to a Chain Saw and the number is ‘012’)**

-in main(), prompt the user if he/she wants to start program or not

-if user want to start, call the main menu.

-Based on the user selection,

---if user select 1 from main menu then prompt with Customer Menu:

------------in Customer menu, if user select 1 then proceed to create this customer by calling the correct function.

------------in customer menu, if user select 2 then proceed to prompt user to provide an ID and then search for it and if found display this customer data.

------------in customer menu, if user select 3 then go back to main menu

---if user select 2 from main menu then prompt with Tool Menu:

------------in Tool menu, if user select 1 then proceed to create a Tool by calling the correct function.

------------in Tool menu, if user select 2 then proceed to prompt user to provide an ID and then search for it and if found display this Tool data.

------------in Tool menu, if user select 3 then go back to main menu

---if user select 3 from main menu then prompt with Transaction Menu:

-------------in Transaction menu, if user select 1 then proceed to initiate a transaction by :

------------------------------prompt user to enter a Customer ID, search for it and if found prompt user with a list of available tools to choose from by ID, search for the tool ID and if found then prompt the user to select between full day or half day rental, then call the correct function that will populate this transaction by Customer ID, Tool ID, Code for Rental Period (‘F’ or ‘H’) , and the Rental Fees.

-------------------------------If Customer ID for a transaction is not found then instruct the user to go back and create a new account for this customer.

-----------------in Transaction menu, if user select 2, then ask user if he want to do a search by Customer ID or Tool ID and proceed accordingly. Warning! If user select a search by Customer ID then you must retrieve all transactions for this customer (a customer can rent one or more tools)…display the corresponding record by showing all data about the customer, and the tool.

Note that Customer ID and Tool ID make up what we call a composite key…both together refer to one and only one transaction (extra credit : can you encode a transaction ID made up of both Cust ID and Trans ID?...may be required in Phase II)

------------------in Transaction menu, if user select 3, then call the correct function that you coded earlie class, display everything about a transaction and related customers and tools. Use a tabular format with the related headings (the format is up to you).

Phase II and XC: partial specs.

-all transaction must be time stamped

-if a tool is not returned either 6 hours max past its rental time for a half day rental or 12 hours max past its rental time for a full day rental, then the fees will be doubled.

-all data for all entities must be saved to and retrieved from a binary file.

Past Phase II and for XC:

Recode phase II using C++ and an OOP approach.

(try simulating time speed via a random function that would cause some rental to be over due).

🡺some of these may turn to be for extra credits if time does not allow it.

A thread for this project exist under Discussions….take advantage of it to discuss the specs with your classmates.