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

**Assignment #8**

Due Date: Nov 1, 2017 at 9:25am (MW class), Nov 2, 2017 at 9:25am (TR class)

Purpose: Practice designing a software solution

Effort: Individual

Points: 100

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

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

**This is design assignment, not a coding assignment.**

**Assignment Description**

Assume there is a company with an online presence. Each day all customer visits to the company’s website are recorded in a daily log file. Each entry in the daily log file contains a customer ID, with one entry per line. A customer may visit the website several times on one day; each visit will create an entry in the log file. Given two daily log files, for different days, produce a list of the customers who visited the website on both days. This list should be written to a new file containing the IDs of those returning customers. It should contain no duplicate customer IDs.

**Specifications**

1. Create a design document called **Assignment 8 Design** or whatever you’d like to call it.
2. At a high level, there are several steps to take a problem statement to a software solution. Include in your design the items in steps 2 – step 4:

**Step 1: Problem Statement**

* + - * 1. The first step is to understand and simplify the problem that needs to be solved.
        2. A simple feature list can help in developing understanding. For example,

The system must support files

The system reads from files and creates a new file

**Step 2: Conceptual Model (include in your design)**

* + Create a very simple conceptual model from the problem statement. A conceptual model captures the ideas in a problem represented by top-level **concepts** and **actions**.
  + To create a conceptual model from the problem statement:
  + First list the important concepts in the problem description.
    - Things, people, time intervals, events, etc.
  + Second list the actions that must be performed to solve the problem.
    1. Where possible use the concept names you just listed.
    2. Look at the verbs in the problem description - explicit or implicit
    3. Break the problem down into smaller pieces of functionality while keeping your design at a high level:
       1. **Do:**
          1. Use terms like *find, read, write, store, match, sort, compare, unique, duplicate*
       2. **Do not:** 
          1. Write C syntax
          2. Worry about C data constructs

**Examples:**

*Read customer ID from daily log file*

*Write customer ID to returning customers file*

**Step 3: Pictures/Diagrams (include in your design)**

* + A picture is worth a thousand words! Pictures help in understanding problems!
    - In assignment 6, we had data that needed to be extracted from a file and placed into arrays. The arrays then needed to be searched and updated. This was a perfect situation for a picture.
  + Draw a picture of the problem.
    - Include and label how data flows through the system.
    - Where appropriate, label the actions that occur in the data flow.

**Step 4: Pseudocode (include in your design)**

* + Create a top-level description of the problem from the conceptual model. Start the process of creating the skeleton for the program where each action becomes a function.
    - Write pseudo code or draw a flowchart for the top level (this turns into main)
    - Write pseudo code or draw a flowchart for the body of each function.
    - The pseudocode or flowchart must be:
      * Describe how the function solves that specific part of the problem.
      * Contain the solution for the action the function performs.
      * Deal with happy path as well as the error paths.

Understanding problem

Simplification of problem statement

Top level concepts and actions

Design – Pseudocode, Diagrams

Source code!