Project 9

C++ Structures

Project 9 Overview - 1

- Brief Description:
  - You are writing the subscription renewal system for a magazine
  - Information for each subscriber is kept in a file
  - The information for one subscriber is called a record, and the input file may contain several records or no records
  - After reading each record, a test is made to determine if the subscribers' subscription has expired
  - If a subscribers subscription has not expired, no action is necessary and the next record is obtained
  - If a subscribers' subscription has expired, information about the subscription is to be written into an output file before the next record is obtained.

Project 9 Overview - 2

- Project 9 introduces the student to the use of ***structures*** in programming.
- In addition to ***structures***, this assignment utilizes several concepts we have previously covered including
  - Input from a file
  - Output to a file
  - Functions (value returning and void)
  - IF statements
  - WHILE loops (or some other form of looping)
- Run the sample solution using the input files contained in P9_in.zip

/home/work/cpe211/Executables/Project_09/Project_09_solution

Project 9 Constraints - 1

- Use only material from Chapters 1-10 of your textbook
  - **We want to familiarize you with the concepts in these chapters prior to the next exam**
- The use of any of the following specific concepts are ***not*** allowed on Project 9 and will result in ***zero (0) credit*** on this assignment
  - Global variables
  - Arrays
  - Recursive function calls (a function invoking itself)
- ***Make your program's output match that of the provided sample solution***
- ***The comparison script can be run as***

/home/work/cpe211data/Project_09/CompareSolution.bash Project_09.cpp

Project 9 Constraints - 2

- Your program must utilize ***at least Four non-trivial functions*** in addition to your main() function to receive full credit
- ***These four functions are required***
  - ***A function that opens the input file and includes error-handling logic***
  - ***A function that opens the output file and includes error-handling logic***
  - ***A function that inputs one complete record at a time from the input file and stores the information in the appropriate structure variable fields (member names) – this function cannot have an ofstream parameter***
  - ***A function that writes a subscriber record to the output file if and only if that subscribers subscription has expired . This function cannot have an ifstream parameter***

Project 9 Constraints - 3

- *Functions can have at most 3 parameters and only one of the parameters can be a structure(so it should be the top level structure)*
- **The program can open the input file one time only**
- ***All user defined functions are called from main only.*** *No user defined function can call another user defined function.*
- ***Output written to the output file must occur in the function for writing the output and these statements cannot be in the function that reads the information.***

Project 9 Structures - 1

- Five structures that you must declare and use in your programs

- Members of the Top-Level Structure
  - Subscriber name (declared as a **Subscriber Name** structure)
  - Address (declared as an **Address** structure)
  - Renewal information (declared as a **Renewal Information** structure)

Project 9 Structures - 2

- Members of the **Subscriber Name** Structure
  - first name (a string)
  - last name (a string)
  - customer id (an integer)

- Members of the **Address** Structure
  - street number and name (a string)
  - city (a string)
  - state (a string)
  - zip_code (an integer)

Project 9 Structures - 3

- Members of the **Date** Structure

– month	(a string)
– day	(an integer)
– year	(an integer)

- Members of the **Renewal Information** Structure

– number of months left	(an integer)
– last notice sent	(a **Date** structure)

Project 9 Structures - 4

- Structure declarations
  - The order in which you declare the structures matters
  - Any nested structure (sub structure) must be declared before the associated top-level structure
    - **Date** Structure must be declared before **Renewal Information** Structure
    - **Subscriber Name** Structure, **Address** Structure, Date Structure, and **Renewal Information** Structure must be declared before the Top-Level Structure
  - All structure type declarations should appear between the ***using namespace std;*** statement and your function prototypes
    - ***Remember, do not declare any struct variables here since that would make them global variables – declare just the structure data types here***

Project 9 Input File Format

- **You may assume that all fields are present for each record, and that the information in those fields is of the appropriate data type.** ***Note, comments shown are not present in the actual input file.***

```
Joe                 // First name
Smith               // Last Name
14235               // Customer id
103 Abbey Way       // Street Number and Name (must be read with getline)
Huntsville          // City (must be read with getline)
Alabama             // State (must be read with getline)
35789               // Zipcode
2                   // # of months remaining
March               // Last renewal notice month
10                  // Last renewal notice day
2002                // Last renewal notice year
```

Project 9 Hints - 1

- The function **main()** should contain the loop that controls the reading of the records
  - In main(), two counters are to be used to keep track of (1) the number of subscribers present in the file and (2) the number of subscribers that have had their subscriptions expire
  - As each record is being read, output which customer ID is being processed – see the sample solution.
  - After the last subscriber information has been processed, a message is to be written to the terminal indicating how many subscribers were in the file and how many subscriptions have expired

Project 9 Hints - 2

- The function that opens the file for input (or output) should contain appropriate error handling
  - If the user-specified file does not open, this function should use a **LOOP** that will repeatedly
    - output an error message
    - close and clear the file stream
    - **`inputStreamVariable.clear();`**
    - reprompt the user and input their next file selection
    - attempt to open the next selection
  - until the user-specified file opens successfully or until the user types Control-C to exit the program. ***Remember – Recursion is not allowed!!***
  - Read the Project description