Programming Assignment 2

Posted: Monday, June 25
Sprint 1: June 25 - July 2
Sprint 2: July 2 - July 11
Due Date: Wednesday, July 11
DDD: Wednesday, July 18

Statement of Work
Programming Assignment 2

1.0 Overview

Smith, Smithe, Smyth, Smythe, and Jones is a wholesale computer equipment supplier which sells computers, computer parts, and computer perpherials to a large number of retail stores throughout the United States. The company would like to create a database of all its traveling sales representatives. This database must be capable of maintaining the employee ID, employee name, department, and annual salary of each sales rep. In addition it must also maintain a list of stores and their locations for each employee. The second phase of development for this database will be to create the list of customer stores for each employee in the database.

2.0 Requirements

The student shall define, develop, document, prototype, test, and modify as required the software system.

2.0.1 This software system shall consist of three source files (.cpp) and three header files (.h) defining three classes. The first class, which was developed in phase one is the EmployeeRecord class. It will be modified to hold a list of stores served by an employee. The second class, which shall be called **Store** shall be used to store information on a single customer store. The third class, which shall be called **CustomerList** shall define and maintain an ordered linked list of instances of the class **Store**.

2.0.2 The class **EmployeeRecord**, which was defined in Phase 1 shall be modified in order to maintain a list of stores served by the employee.

2.0.2.1 A pointer to a class of type **CustomerList** called **m_pCustomerList** shall be added to the EmployeeRecord class. This pointer shall be private.

2.0.2.2 The constructors of the EmployeeRecord class shall be modified so that each creates an instance of a CustomerList object and sets the pointer m_pCustomerList pointing to this object. The destructor of the EmployeeRecord class shall be modified to delete this instance of CustomerList.

2.0.2.3 The function **CustomerList *getCustomerList()** shall be added to the EmployeeRecord class. This function shall return the pointer to the EmployeeRecord's CustomerList object.

2.0.2.4 The function **getDept()** shall be modified to take no arguments but return the int value of m_iDeptID.

2.0.2.5 The function **getSalary()** shall be modified to take no arguments but return the double value of m_dSalary.

2.0.3 The class **Store** shall maintain information on a single customer store for the employee. This class shall hold the store ID number, name, address, city, state, and zip code. It will contain get and set functions for each of these items. It will also contain a pointer called **m_pNext** which will enable a linked list of Store objects to be created and maintained by the class CustomerList. **The header file and the implementation file for this class will be provided by the instructor.**

2.0.4 The class **CustomerList** shall create and maintain an **ordered linked list** of Store objects arranged by Store ID.

2.0.4.1 The CustomerList class shall contain the following private variable: **Store *m_pHead** which shall point to the first Store instance in the list.

2.0.4.2 The CustomerList class shall contain the following public functions: **bool addStore(Store *s), Store *removeStore(int ID), Store *getStore(int ID), bool updateStore(int ID, char *name, char *addr, char *city,**

char *st, char *zip), and **void printStoresInfo()**.

2.0.4.2.1 bool addStore(Store *s)--This function shall take a pointer to a Store object which already contains all data on a store. It shall insert the Store object into the linked list **in order, sorted by the store ID**. It shall return TRUE if the Store was successfully added to the list.

2.0.4.2.2 Store *removeStore(int ID)--This function shall take an integer store ID as an argument. It shall search the list, locate the Store object with that ID if one is present, remove it from the list and return the Store object. The function shall return NULL if it failed to find the Store in the list.

2.0.4.2.3 Store *getStore(int ID)--This function shall take an integer store ID. It shall search the list, locate the Store object, if present, and return a pointer to the Store object. It shall return NULL if the Store was not found in the list.

2.0.4.2.4 bool updateStore(int ID, char *name, char *addr, char *city, char *st, char *zip)--Function shall take a list of arguments defining changes in the store data. The first argument gives the store ID. The remaining arguments are char arrays containing the new data on that store. The function will search the list and locate the store then update all data for that store. The function will return TRUE if it successfully updated the data or FALSE if it failed to find the store.

2.0.4.2.5 void printStoresInfo()--Function shall print all data on each store in the list.

3.0 Deliverables

These products shall be delivered to the instructor electronically via e-mail as specified below.

3.1 Sprint Report -- The student shall provide a filled out Sprint Report form for instructor approval NLT (Not Later Than) Wednesday, July 11.

3.2 Program source files -- The student shall provide fully tested electronic copies of the .cpp and .h files. These files must be submitted to the instructor via e-mail. The files shall be delivered NLT Wednesday, July 11.

4.0 Period of Performance

The period of performance of this assignment is 28 days from the date of assignment. The DDD for source code is Wednesday, July 18.

Source code for the Store class, and a demonstration executable of this program can be found in a .zip file on the Downloads page..