CMPS 260
Spring 2005
Project #4
2005.03.14b

Date Assigned: **Monday, March 14, 2005**
Due Date: **10:00 PM, Sunday, April 3, 2005**

The coded solution to the following problem is to be done by you and only you. You may ask help from the class teaching assistants and the instructors, but you may not ask for help on this project from anyone else. You may use your notes, C++ texts, on-line tutorials, etc., but the code must be your own.

1. Project Description:

The Personnel Department of General Forge and Foundry, Inc. is converting to computerized records of its employees. The good news is that a file of existing employees already exists, as does a file a file of job descriptions. The bad news is that you have to write the program.

The information that is stored in the file about each employee (payrole.dat) is employee id number, salary and job code. Here is some example data from the file:

```
1000        // employee id number
20886       // salary
3           // job code
1001
36915
7
1002
28335
3
...
```

The job descriptions are stored in a file (jobs.dat), which is as follows:

```
tinker
tailor
soldier
spy
doctor
lawyer
receptionist
manager
personnel
accounting
```

The job descriptions are in the order of the job codes. I. e. tinker is job code 0, tailor is job code 1, etc.

The program must load the data from the files at the start of each run and save the employee data back to its file when the program exits. The program must also provide the accounting department with the following operations:

- display the information on all employees with column headers and pauses every 20 employees; the job description is output instead of the job code
- based on a user input employee id, display the information on a specific employee; the job description is output instead of the job code
- based on a user input employee id, allow the salary to be changed of a specific employee
- allow the job coded to be changed of a specific employee based on a user input employee id; the jobs descriptions must be listed but the user will input the job code
- delete a specific employee based on a user input employee id
- add a new employee; the id used must be unique

2. Design Requirements:

(a) Employee data is to be stored in an object of the template version of class arrayListType. There will be at most 200 employees.

(b) Design a class to model the information about one employee, including id, salary and job code. The class must have constructors and methods such that each value can be set and retrieved. The class must be used as the data type of the element of class arrayListType, so it must have overloaded methods for all relational operators, the assignment operator and friend functions for overloading the input and output stream operators.

(c) The job descriptions can be stored in a string array or in an object of class arrayListType. The job descriptions are to be looked up as needed (as for example: displaying). There are 10 job types. Job codes number from 0 to 9.

3. Existing Files:

The files jobs.dat and payrole.dat are available on the UCS network at /w1/cs260x (where x is your section number). The files arrayListType.h and arrayListType.cpp are available on the UCS network at / w1/cs260x/template (where x is your section number)

4. Additional Requirements:

- You are responsible for following the requirements as given in documents reachable from the class web site via the **Minimum Documentation** and **Naming Conventions** links. Methods in your class definitions must has accompanying pre-condition and post-condition documentation. (See the link **Additional Documentation** on the class web site.)

- A makefile is to be created for and submitted with this assignment.

- Each class design is to be placed in a file named for the class and ending in ".h" (a header file). Each class implementation is to be placed in a file named the same as the file of the design, but ending in ".cpp". Each header file is to include wrappers to prevent unnecessary recompilation

- Your program may have global constants but may not have global variables.

- Your program solution must use good programing style.

5. Submitting: You are responsible for submitting your work both electronically and via a hard copy.

- Instructions for submitting an electronic copy of your project can be found on the class web site, via the **Submitting Your Work** link. When prompted, name your project "proj4".

- A hard copy of the code in your project is due the school day after you submit electronically. Place the print out of your code in a manila folder, write your name, clid and section on the folder, then bring the folder to class or your instructor's or TA's office.