**Overview**

We will practice C++ in Nachos by **implementing Employee Management Software for the small company XYZ**.

If there is a file, **employee.dat**, read all employee records, create Employee objects and make a list. After reading employee.dat, show the following menu repeatedly until menu 7 (Exit) is selected.

1. Enter new record
2. Display all employees
3. Search employee(s)
4. Update employee info
5. Delete employee info
6. Schedule weekly jobs
7. Exit

First, enter a new record of an employee with his/her unique ID. If the employee is the first one (there is no existing employee list), his/her ID is starting with 8001. If not, the new employee’s ID will be assigned to the highest ID of existing ones + 1. The unique ID is automatically assigned by the system, not user input. Enter all other employee info manually.

Second, display all employee records to screen. The list should be sorted by department and position. There should be at least 3 departments and 4 position ranks.

Third, search employee(s) with either department or ID. You can display a message for user input. If department is entered, display all members of the department. If employee ID is entered, display the record of the employee.

Fourth, update employee record with employee ID. All information except ID can be updated.

Fifth, delete employee record with employee ID. This process **needs a confirmation to delete** the record of selected employee.

Sixth, schedule weekly jobs. There is a job queue for all employees. Each employee has a certain amount of task to do, which should take a certain amount of time. The time should be **randomly generated 20 to 40 hours**. With this time, each employee will receive his/her **weekly paycheck** (time \* hourly pay rate). HR needs to know the total paycheck amount for the week and personal paycheck for each employee.

Seventh, exit the program. All record of current employees must be saved to employee.dat before exiting.

**Output:**

For each menu selection, you should print out appropriate messages interactively.

* For menu 6, after people get their task,
  + For each employee (must be sorted by paycheck amount in descending order)
    - Order # (starting from 1)
    - Who is working (person’s ID and name)
    - How long the person has worked for the week
    - How much the person will be paid
  + Total amount of paycheck for the week

**Implementation Requirements:**

* Define a class for Employee, which contains all the information you need for an employee, and use it to create employees (Employee objects).
  + Implement the Employee class in the new files (declaration in .h file and definition in .cc file)
  + Required Employee record info
    - Name
    - Unique ID
    - Position
    - Department
    - Hourly pay rate
  + The member variables have to be private. Implement public getter/setter functions for accessing these variables
  + Change code/build.linux/Makefile to include the new files to the NachOS compilation process
* Call rand() function to randomly generate a number.
* Implement the list of employees using the functions defined in code/lib/list.h and list.cc
  + Read all the codes and comments in list.h, list.cc, and code/threads/threadtest.cc
  + You need to sort all employees differently for each menu selection. For list manipulations, the existing functions in list.h/cc should be enough.
  + Hint: ThreadTest() function is called by main()
* Use List::Apply() function for printing the information.
  + You need to define your own print functions in threadtest.cc, for example:  
    void PrintRecord(Employee rec);

Void PrintPaycheck(Employee working);

* + List::Apply() will apply the function PrintRecord to every element in the list.
* The entire interactive process should be done in threadtest.cc.
* Your implementation must be incorporated and compiled with NachOS under thread.