CMSC 127

File Processing and Database Systems

Exercise 6: File Processing

Objectives

At the end of this exercise, you should be able to:

- construct a simple working database management system using a file; and
- parse complex terms in sequential files.

Lore

The municipality of Rosario in Batangas has tasked you to develop an employee management system. But since the local government unit (LGU) has no money to deploy and maintain one, you were asked to create a command-line based program for the meantime.

A key informant interview was conducted in the town hall of Rosario with the project interviewer and the municipal administrator which is one of the key users of the program. On the said interview, the following details were provided for and given by the administrator:

- You are to create a C-based program to manage the employee records of the town.
- Each employee has the following characteristics and attributes:
    - a unique seven-digit employee number;
    - a name, consisting of first, middle and last names (assume string length of 30 characters);
    - the amount of his or her monthly salary in Philippine pesos;
    - his or her appointment date in the job (MM/DD/YYYY);
    - his or her birthdate (MM/DD/YYYY); and
    - his or her position in the government service.

Technical Requirements

Since in CMSC 123, the deletion algorithm in AVL trees was not taught, you are to implement this system using only a binary search tree as a primary data structure.

The system that is required from you needs to have the following functionalities, as well as its corresponding input error-checking and input confirmation capabilities.

- Add employees
    - Ask the number of employees to be added.
    - Add the corresponding number of employees in the order of the requirements stated above.
    - If there are duplicates in the employee number, prompt for error and ask for re-entry.
- Search employee
    - Search the employee by employee number.
    - Returns the complete employee information in list form if found.
    - Prompt an error statement otherwise.
- Edit employee

    - Search the employee by employee number.
    - If found, let the user edit every information except for the employee number.
    - Prompt an error statement otherwise.
- Delete employee
    - Search the employee by employee number.
    - If found, delete the employee on the database.
    - Prompt an error statement otherwise.
- List all employees
    - List the employees by employee number in ascending order in tabular form.
    - List only the employee number, name, position and salary **per annum**.
- Delete all employees
    - Delete all employees in the system.
- Write employees in file
    - Write all of the information of the employees in a variable text file, with the filename being provided by the user.
    - The file format will be explained later.
- Load employees from file
    - Load all of the information of the employees from a file named “employees.in”.
    - The file format will be explained later.

Submission Details

DEADLINE	11:59:59pm, one week after
LATE HOURS	Deduction of one point per one late hour
E-MAIL	`exer.receiver@gmail.com`
SUBJECT	`[CMSC 127 <lab_section>] Exer 6` e.g. `[CMSC 127 ST-1L] Exer 6`
FILE NAMES	`exer6.c` `exer6.h`

Bonuses

1. Enable the string typed attributes to have infinite length. [3]
2. Have proper code reuse, structured walkthrough and modular design. [3]

Rubric

100% (25)	Code works in all (100%) of the test cases and has proper documentation.
70% (17.5)	Code works in most (70%) of the test cases and has proper documentation. Code works in all (100%) of the test cases and has no proper documentation.
40% (10)	Code works in some (40%) of the test cases and has or has no proper

	documentation.
10% (2.5)	Code works in 10% of the test cases and has or has no proper documentation. Code compiles and runs.
0% (0)	Student cheated. Student did not follow the submission guidelines. Student did not attach files to the e-mail.

File Format

DELIMITER	*
ORDER	In order as seen in the lore.
NOTE	For uniformity, use camel case with the first letter of the word capitalized.
EXAMPLE	0854097*Rick Jason*Gallarde*Obrero*23044.00*06*02*2012*11*11*1992*Instructor 1