UNIVERSITY OF TECHNOLOGY, JAMAICA

School of Computing and Information Technology

Data Structures Project

D. White / T. Edwards / L. Henry / S. McNamarah / R. Kotha

Group Assignment (3 or 4 persons to each group)

Given Week of February 8, 2010 **Due Week of March 8, 2010**

Software Solutions Now (SSN) is a software development company specializing in the development of new programs and the customization of off the shelf applications. SSN was started as a small freelancing venture by two friends, the small venture as grown into a thriving business with over one hundred employees.

Over the years, the company has maintained a simple payroll system in a spreadsheet application, which was sufficient based on the number of employees. Now that the business has grown, the system is no longer suitable, and as such the manager of the Accounts department has requested that a new system be developed.

The manager provided the following information to the Software Engineering Department, to aid in the development of the SSN Payroll Management System (PMS).

Each employee is assigned to a department and is paid based on the rates setup for each department; these rates are stored in a department rates file (the structure of the rates file is shown below). These rates are hourly rates. The overtime rate is applied once an employee works in excess of 40 hours.

Department Rates File (Sample Data)

Dept. Code	Dept. Name	Regular Rate $	Overtime Rate $
1001	Human Resource Management	85.50	10.00

A separate employee payroll file is maintained that stores the data on each employee (the structure of the file is shown below), both files (**Rates File & Payroll File**) are tab delimited text files, with the respective column headings.

NB: Sample Rates and Payroll files will be provided, for development purposes.

Employee Payroll File (Sample Data)

ID. No	First Name	Last Name	Dept. Code	Position	Hours Worked
2201	Michael	Hall	1001	Manager	28.50

The processing of the payroll leads to the generation of a processed payroll file (the structure of the file is shown below).

Processed Payroll File (Sample Data)

ID. No	First Name	Last Name	Dept. Code	Position	Hours Worked	Regular Pay	Overtime Pay	Gross Pay
2201	Michael	Hall	1001	Manager	28.50	2436.75	0.00	2436.75

NB: None functional projects will automatically FAIL

Program Requirements:

1. Perform an object oriented analysis on the proposed SSN Payroll Management System (PMS) described above, and document the development of the system solution using pseudo code specification and appropriate diagrams. Each data structure used should be specified using pseudo code, the sorting method used should also be specified using pseudo code.

2. Using Visual C++, implement the SSN Payroll Management System (PMS) using the projects feature of the IDE.

3. The user shall maintain the Department Rates data for each department after loading the data from the text file, via a menu with the options:
    - **Add**: Allows the user to add a new department rates record to the system.
    - **Update**: Allows the user to update an existing department record.
    - **View**: Allows the user to view a single department record.
    - **View All**: Allows the user to view all department records.

4. The user shall maintain the data for each employee after loading the data from the text file, via a menu with the options:
    - **Add**: Allows the user to add a new employee payroll record to the system.
    - **Update**: Allows the user to update an existing employee payroll record.
    - **View**: Allows the user to view a single employee payroll record.
    - **View Sorted**:
        - **ID. No**: Allows the user to view all employee payroll records sorted by ID. No.
        - **Last Name**: Allows the user to view all employee payroll records sorted by Last Name.
        - **Dept. Code**: Allows the user to view all employee payroll records sorted by Dept. Code.
        - **Position**: Allows the user to view all employee payroll records sorted by Position.
        - **Hours Worked**: Allows the user to view all employee payroll records sorted by Hours Worked.
    - **Delete**: Allows the user to delete an employee payroll record.

5. The user shall process the payroll after loading the data from the text files, via a menu with the options:
    - **Process Payroll**: Calculates payroll and generates Processed Payroll File.
    - **View Payroll**: Allows the user to view a single employee payroll record.
    - **View Sorted Payroll**:
        - **ID. No**: Allows the user to view all employee payroll records sorted by ID. No.
        - **Last Name**: Allows the user to view all employee payroll records sorted by Last Name.
        - **Dept. Code**: Allows the user to view all employee payroll records sorted by Dept. Code.
        - **Position**: Allows the user to view all employee payroll records sorted by Position.

6. When the exit option is selected the application should close.

7. All committed changes made during the execution of the program, should be stored and used to update the relevant files, when the application terminates.

<u>NB: None functional projects will automatically FAIL</u>

Deliverables Required:
Submit the complete source code, executable file, data file and documentation including programme listing, screenshots and sample run on CD as well as hardcopy to Student Affairs by **2010 March 12** (four weeks after issue date).

Submit along with the printed documentation, completed Authorship forms, one for each member of the group and a project report detailing the contribution of each member and any challenges encountered.

The group shall attend an interview during which the project will be presented and marks awarded, to those in attendance. **Marks will not be awarded if the interview is not done.**

Marks will be awarded as follows:

- Data Structures (15 %)
    - Use of appropriate structures (e.g. Classes)
    - Use of Linked Lists

- Functionality (40%)
    - Proper implementation and use of classes
    - Proper implementation and use of Linked Lists
    - Demonstration of Searching and Sorting
    - Retrieving and Storing information in a text file
    - Robustness
    - Correctness

- Documentation (20%)
    - Internal (Program)
        - Comments, indentation, naming convention, white space
    - External (Printed document)
        - Pseudo codes, formatting & neatness, sample run, screenshots, project report, authorship forms

- User Interface (5%)
- Ease of Use (5%)
- Modular Design (10%)
- Originality & Ingenuity (5%)

Marks will be subtracted for late assignments at a rate of 5% per day. Assignments more than one week late will not be accepted.

EXTRA CREDIT:
+ 5%: Awarded for the extended keyboard keys (F1 – F12, Home, Escape, Arrows Keys etc.) to improve the feel (how the user interacts with) the interface.

+ 5%: Awarded for the use of colour, extended ASCII codes, or graphics to improve the look of the user interface.

+ 10%: Awarded for allowing the user to print the Processed Payroll File as a properly formatted report.

NB: Extra credit features should only be done if the basic solution is working **i.e. basic requirements MUST NOT be sacrificed for the extra credit features**.

A program that does not satisfy the basic requirements will not be considered for extra credits even if they are properly implemented.

The maximum extra credit is 20%.

NB: None functional projects will automatically FAIL