

# **CS4013 Object Oriented Development Project**

## **Objectives:**

- Apply an object-oriented approach to the design and development of a software application in Java.
- Apply the concepts and techniques introduced in lectures, tutorials and labs to the design and development of a small software application.

## **Payroll System**

### **System Requirements:**

You are required to build a Payroll system for staff. There are similar salary scales for hourly paid employees. For example, there may be different scales such as Lecture, Tutorial, Lab, Demo, Exam Invigilator, Exam Supervisor etc... Each of these scales would also have several points. Part-time employees must submit a pay claim form by the second Friday of a month in order to be paid that month. The payroll system should generate pay slips for all full-time staff and hourly paid staff (with current claims) on the 25<sup>th</sup> day of each month.

When calculating net pay for a payslip, certain deductions must be made. These include deductions for Health Insurance, Union Fees, and for the various taxes (e.g. PRSI, USC, Income Tax).

In October each year, full-time staff are moved to the next point on their salary scale (if they are not already at the top of that scale. From time to time an employee may be promoted to the next salary scale within their professional category (Academic/Administrative/IT/Technical etc..). The point to which a staff member is promoted to the new salary scale is dependent on the number of years spent at the top of the previous scale.

All relevant data associated with the payroll system (such as employee, payroll, payslip details, including historic payslips) and any other files you deem appropriate for this application should be stored as csv files.

The application has three user types. An employee can log in to the system, see their details, and view their most recent or historical payslips. An admin user can log in and can add a new employee to the system. A human resources user can log in to implement the promotion functionality for full-time staff. In such cases the staff member should be asked to confirm the promotional changes being applied. A command line interface (CLI) should be included to facilitate this interaction between users and the application. The CLI should be included with the expectation that a Graphical User Interface (GUI) may be required in the future. In other words, the separation between the text-based user interface should be well-defined to allow a graphical user interface to be substituted easily.

## **Suggestions**

- In terms of managing the project, consider adopting some of the ideas from an Agile approach such as Scrum...for example such an approach advocates daily meetings. In the context of this project and your work on other modules, maybe you could agree to meet every 2-3 days where it is made clear what each team member will work on. See the contributions file where details of such meetings should be recorded.

## **Project Deliverables**

This project must be implemented in the Java programming language. As can be seen from the deliverables below this project is not just an implementation project, it also involves design and documentation.

The following are the deliverables for the project:

- A document outlining the Class Responsibility Collaboration (CRC) cards. These can be created in a word document.
- A UML class diagram showing the relationships between the classes. This should include the class names (no details of attributes/constructors/methods required). It must show the relationships between the classes (class inheritance, interface inheritance, aggregation, dependency as appropriate) and utilising the notation presented in lecture slides (please refer to the slides). This UML diagram must be generated manually and drawn using a tool like draw.io.
- Documentation for the software generated using the javadoc utility and a help file describing briefly how to run the application.
- The source code for the system where each Java class/interface is stored in a separate file. Use packages to group files as appropriate.
- Any csv files that are required by the system should also be included. These files must be populated with data (e.g. sample data used to test the application).
- The use of a Github repository for source code version management.

## **Rules and Regulations**

- You must undertake this assignment on a group basis. Strictly groups with max size 4.
- Lab interviews may be undertaken before the deadline to establish your progress towards completing this assignment. Marks may be awarded for this purpose. Non-attendance at your specified lab may result in you forfeiting these marks.
- Accidental loss of work will not be accepted as an excuse.