1st SIT COURSEWORK QUESTION PAPER: **Year Long Spring 2019**

Module Code:	**CS4001NA**
Module Title:	**Programming**
Module Leader:	**Dhruba Sen** (Islington College)

Coursework Type:	**Individual** **Teacher hiring system (GUI)**
Coursework Weight:	This coursework accounts for **30%** of your total module grades.
Submission Date:	**24th Week**
When Coursework is given out:	**20th Week**
Submission Instructions:	Submit the following to Islington College RTE department before the due date: • **A report in PDF format and zip file which includes program file** • **File should be in .java format**
Warning:	London Metropolitan University and Islington College takes Plagiarism seriously. Offenders will be dealt with sternly.

Plagiarism Notice

You are reminded that there exist regulations concerning plagiarism.

Extracts from University Regulations on Cheating, Plagiarism and Collusion

Section 2.3: “The following broad types of offence can be identified and are provided as indicative examples …..

(i) Cheating: including copying coursework.
(ii) Falsifying data in experimental results.
(iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.
(iv) Bribery or attempted bribery of a person thought to have some influence on the candidate’s assessment.
(v) Collusion to present joint work as the work solely of one individual.
(vi) Plagiarism, where the work or ideas of another are presented as the candidate’s own.
(vii) Other conduct calculated to secure an advantage on assessment.
(viii) Assisting in any of the above.

Some notes on what this means for students:

(i) Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.
(ii) Taking extracts from published sources without attribution is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. " e = mc2 (Einstein 1905)". A reference section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

Further information in relation to the existing London Metropolitan University regulations concerning plagiarism can be obtained from http://www.londonmet.ac.uk/academic-regulations

Assessment

This assignment will be marked out of 100 and carries 30% of the overall module weighting.
Your .java files and report for this part must be uploaded and submitted by 5pm on Friday of Week 12th. The assignment must be carried out individually so you must not obtain help from anyone other than the module teaching staff. You must not copy code from any source apart from the module core text and the module materials. Collusion, plagiarism (unreferenced copying) and other forms of cheating constitute Academic Misconduct, which can lead to failure of the module and suspension.

Aim

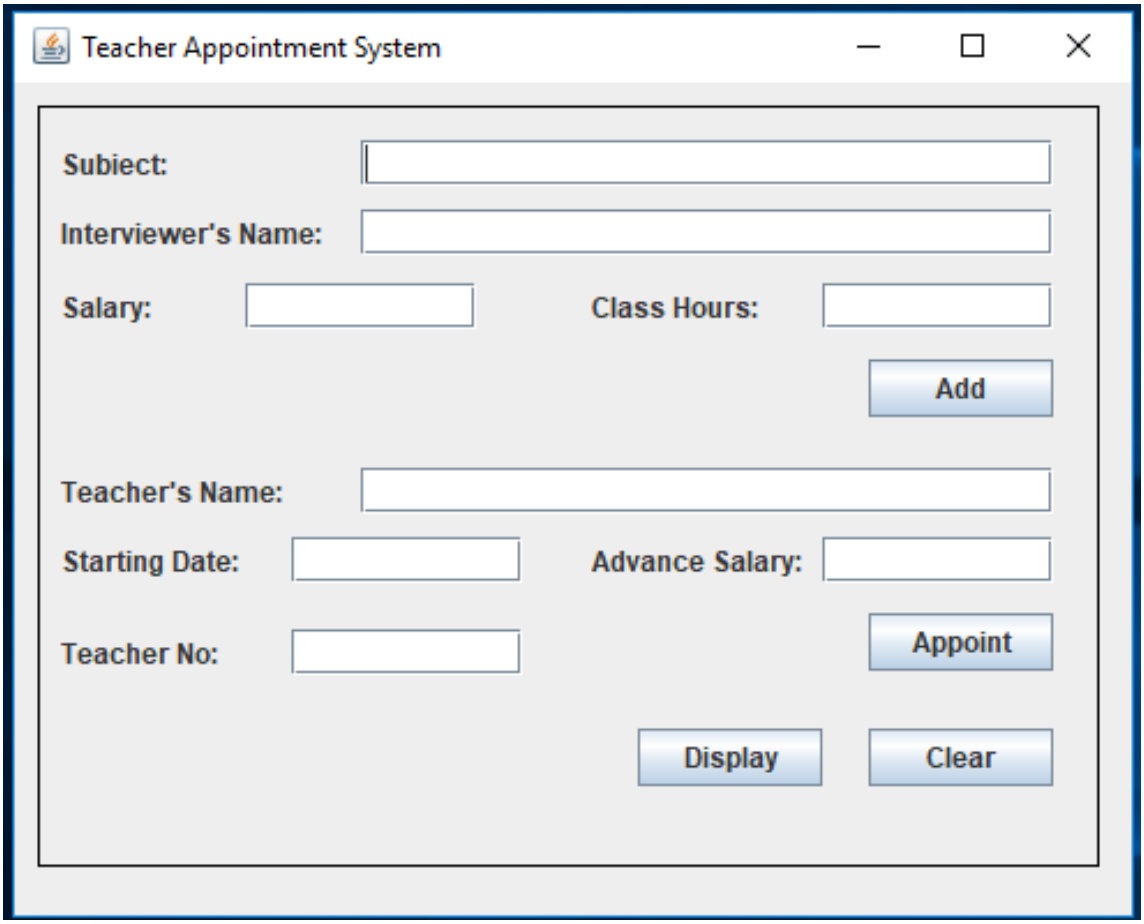
The aim of this assignment is to add a class to the project that you developed for the first part of the coursework to make a graphical user interface (GUI) for a system that stores details of teachers in your teachers list. The class will contain a main method and will be tested using the command prompt. A learning aid about the command prompt is linked from the main index page of the Learning Materials. You will also need to write a report about your program.

Deliverables

Create a new class within the project called **InformaticCollege**. When you are ready to submit your solution, upload your **InformaticCollege.java** file, together with the **Teacher.java**, **Leacturer.java** and **Tutor.java** files from the first part of the coursework (not any other files from the project) together with your report in Microsoft Word format. Your report should be no more than 1000 words in length.

Program (60 marks)

A sample of GUI is shown below:



Your GUI should contain the same components but you are free to use a different layout if you feel that it improves the aesthetics, ease of use etc. The **InformaticCollege** class should store an array list (***not*** an array) of type Teacher to hold the Lecturer and Tutor. There should be text fields for entering:

1. The subject
2. The interviewer name
3. The teacher name
4. The class per day
5. The salary
6. The starting from
7. The academic block number
8. The daily working hour
9. The advance salary
10. The appointed date
11. The termination date
12. The qualification
13. The person who appointed teacher

14. The evaluation period
15. The teacher number (its position in an array list of teachers)

The GUI should have the following buttons:

1) Add Subject for Tutor

When this button is pressed, the input values of the subject, interviewer name and classes per day are used to create a new object of type **Tutor**, which is added to an array list of Teachers.

2) Add Subject for Lecturer

When this button is pressed, the input values of the subject, interviewer name and classes per day are used to create a new object of type **Lecturer**, which is added to the array list of Teachers.

3) Hire Lecturer

The Teacher number, Teacher name, starting date, advance salary and academic block number are entered into the GUI. When this button is pressed, the input value of the Teacher number is checked. If a suitable value has been entered, it is used to get the appropriate lecturer from the array list of teachers. The method to hire the lecturer in the **Lecturer** class is then called.

Hint: An object of Teacher is cast as Lecturer

4) Lecturer contract termination Status

The Teacher number is entered into the GUI. When this button is pressed, the input value of the Teacher number is checked. If a suitable value has been entered, it is used to get the appropriate lecturer from the array list of Teachers. The method to terminate lecturer contract in Lecturer class is then called.

Hint: An object of Teacher is cast as Lecturer

5) Hire Tutor

The teacher number, the teacher name, appointed date, termination date and qualification are entered into the GUI. When this button is pressed, the input value of the teacher number is checked. If a suitable value has been entered, it is used to get the teacher from the array list of teachers. The method to hire tutor in the Tutor class is then called.

Hint: An object of Teacher is cast as Tutor

6) Display All

When this button is pressed, the information relating to properties is displayed:

7) Clear

When this button is pressed, the text is cleared from all text fields.

Hint:

Write methods to return the values of each of the text fields using the ***getText()*** method. For the advance salary and salary get the text from text field, convert it to a whole number and return the whole number. For the academic block number, daily

working hours, evaluation period and the teacher number, get the text from the text field, convert it to a numerical value and return the numerical value.

Additionally, use try & catch blocks to catch any Number Format Exception that might be thrown in converting the string to an integer or double and, if no exception occurs, test that the value is in range. Set the teacher number to -1 if the text input is incorrect in any way and output a suitable error message in a message dialog box.

After calling the necessary method to get the teacher number, Teachers are hired only if the teacher number is not equal to -1.

Marks will be awarded as follows:

1) GUI and main method **[11 marks]**
2) Functionality of buttons **[28 marks]**
3) Reading input, checking input and displaying appropriate message dialogs **[11 marks]**
4) Program style (see http://www.bluej.org/objects-first/styleguide.html) **[10 marks]**

Report (40 marks)

For the report, marks will be awarded as follows:

Your report should describe the process of development of your class with: A class **1)** diagram (of just the InformaticCollege class) **[5 marks]**
2) Pseudocode for each method in the InformaticCollege class **[10 marks]**
3) A short description of what each method in the InformaticCollege class does **[5 marks]**

You should give evidence of the following testing that you carried out on your program:

1) Test that the program can be compiled and run using the command prompt, including a screenshot similar to Figure 1 from the command prompt learning aid. **[2 marks]**
2) Evidence should be shown of:
   Adding subject for tutor list
   Adding subject for lecturer list
   Hiring lecture
   Hiring tutor
   Terminating teachers contact **[5 marks]**
3) Test that appropriate dialog boxes appear when unsuitable values are entered for the teacher number, (include a screenshot of the dialog box, together with a corresponding screenshot of the GUI, showing the values that were entered). **[3 marks]**

The report should contain a section on error detection and error correction where you give examples and evidence of three errors encountered in your

implementation. The errors (syntax and/or runtime) should be distinctive and not of the same type. **[3 marks]**

The report should contain a conclusion, where you evaluate your work, reflecting on what you learnt from the assignment, what difficulties you encountered and how you overcame the difficulties. **[4 marks]**

The report should include a title page (including your name and ID number), a table of contents (with page numbers), and a listing of the code (in an appendix). Marks will also be awarded for the quality of writing and the presentation of the report. **[3 marks]**

Presentation

Note: If student would be unable to defend his/her coursework, s/he might be penalized with 50% of total coursework marks.