LONDON
METROPOLITAN
UNIVERSITY

1st SIT COURSEWORK-2 QUESTION PAPER **Year Long 2023/2024**

Module Code:	**CS4001NI**
Module Title:	**Programming**
Module Leader:	**Sushil Paudel**(Informatics College Pokhara)

Coursework Type:	**Individual**
Coursework Weight:	This coursework accounts for **30%** of your total module grades.
Submission Date:	**Week 24**
When Coursework is given out:	**Week 20**
Submission Instructions:	Submit the following to Informatics College Pokhara RTE department before the due date: **A Report in PDF format and a zip file which includes a BlueJ Project File**
Warning:	London Metropolitan University and Informatics College Pokhara 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 offense. 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 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 RTE deadline.** 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.
The viva will be conducted for this assignment.

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

Aim

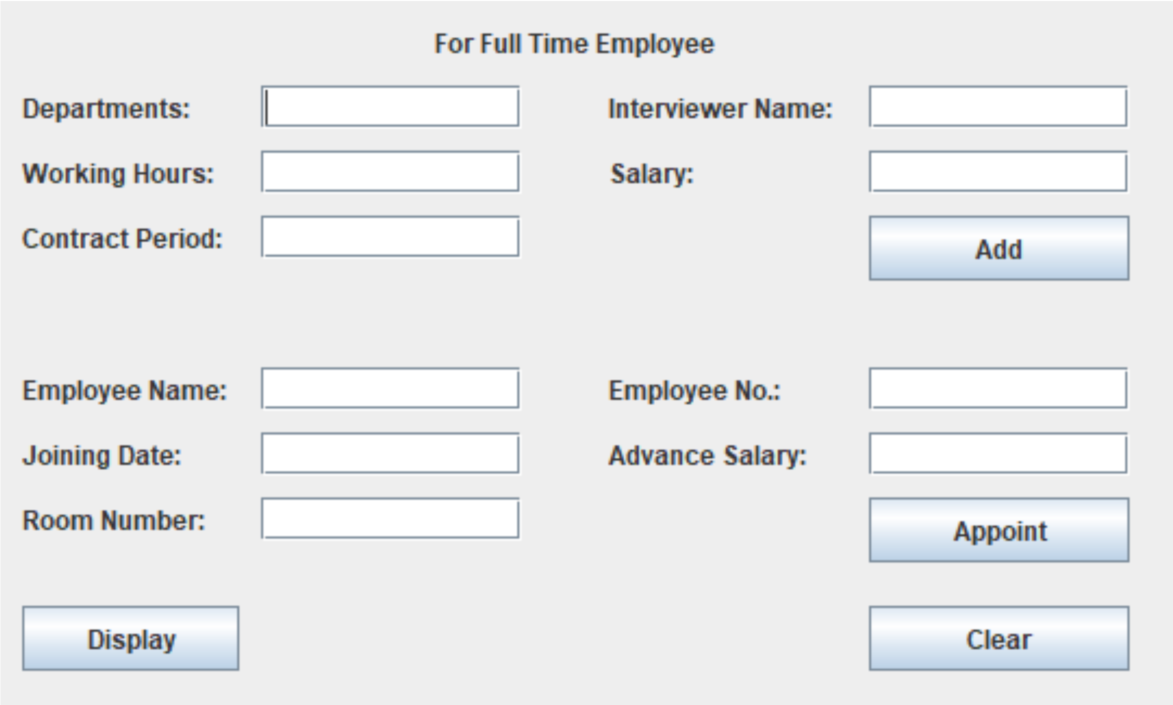
The aim of this assignment is to add a class to the project that you developed for the first part of the coursework to make a graphical user interface (GUI) for a system that stores details of teachers and its details in an ArrayList. The class will contain the main method and will be tested using the command prompt. You will also need to write a report about your program.

Deliverables

Create a new class within the project called **TeacherGUI**. When you are ready to submit your solution, upload your **TeacherGUI.java** file, together with the **Teacher.java**, **Lecturer.java,** and **Tutor.java** files from the first part of the coursework (not any other files from the project) together with your report .pdf format.

Program (57 marks)

A sample of GUI is shown below:



1. 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 **TeacherGUI** class should store an array list (not an array) of the **type Teacher class** to hold **Lecturer** and **Tutor** objects. There should be **text fields** for entering:

i. Teacher id
ii. Teacher name
iii. Address
iv. Working Type
v. Employment Status
vi. Working hours
vii. Department
viii. Years of Experience
ix. Graded Score
x. Salary
xi. Specialization
xii. Academic Qualifications
xiii. Performance Index

2. The GUI should have the following **buttons to**

i. **Add a Lecturer**

When this button is pressed, the input values of teacher id, teacher name, address, working type, employment status, gradedScore, and YearsOfExperience are used to create a new object of type Lecturer which is added to an array list of Teacher class.

ii. **Add a Tutor**

When this button is pressed, the input values of teacher Id, teacher name, address, working type, employment status, working hours, salary, specialization, academic qualifications and performanceIndex are used to create a new object of the type Tutor which is added to an array list of Teacher class.

iii. **Grade the Assignments**

The teacher id, gradedScore, department, and YearsOfExperience are entered in the GUI. When the valid teacher ID is entered in the text box along with gradedScore, department, and YearsOfExperience display an information dialog that displays all data that have been entered. When the grade assignment button is pressed, the input value of teacher ID is compared to the existing teacher ID, and if a valid teacher id has been entered, it is used to grade the assignments from lecturer class. The method to grade assignments from lecturer is called here.

Hint: *An object of Teacher is cast as Lecturer*

iv. **Set the salary of Tutor**

The teacher id, new salary, and new performance Index are entered in the GUI. When the valid teacher ID is entered in the text box, the respective new salary and new performance Index are displayed in the information dialog. When this button is pressed, the input value of teacher ID is compared to the existing teacher ID, and if a valid teacher id has been entered, it is used to set the salary of the tutor. The method to set the salary from the Tutor class is called here.

Hint: *An object of Teacher is cast as Tutor.*

v. **Remove the tutor**

The teacher id is entered in the GUI. When this button is pressed, the input value of teacher ID is compared to the existing teacher ID, and if a valid teacher id has been entered, it is used to remove the tutor. The method to remove the tutor from the Tutor class is called here.

Hint: *An object of Teacher is cast as Tutor.*

Display

When this button is pressed, the information relating to the appropriate class is displayed.

vi. **Clear**

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

Additional Information:
Return the values of each of the text fields using the getText() method. For the number type variable get the text from the text field, convert it to a whole number and return the whole number.

Additionally, use try & catch blocks to catch any Number Format Exception that might be thrown in converting the string to an integer or double. If the text input is incorrect in any way and output a suitable error message in a message dialog box.

Following parts of the programming will be focused mainly to award the Marks

i. GUI and main method **[12 marks]**

ii. Functionality of Buttons **[20 marks]**

iii. Reading input, checking input and displaying appropriate messages for information as well as error dialog **[13 marks]**

iv. Programming Style **[12 marks]**

Report (43 marks)

Your report should describe the process of development of your classes with:

a. A class diagram **[5 marks]**

b. Pseudocode for the class TeacherGUI **[8 marks]**

c. Method description of what each button does **[10 marks]**

d. You should give evidence (through appropriate screenshots) of the following testing that you carried out on your program:

Test 1: Test that the program can be compiled and run using the command prompt, including a screenshot like Figure 1 from the command prompt learning aid. **[2 marks]**

Test 2: Evidences should be shown of: **[2 marks]**

a. Add the Lecturer
b. Add the Tutor
c. Grade Assignments from Lecturer
d. Set the salary
e. Remove the tutor

Test 3: Test that appropriate dialog boxes appear when unsuitable values are entered for the Teacher ID, (include a screenshot of the dialog box, together with a corresponding screenshot of the GUI, showing the values that were entered). **[3 marks]**

e. 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]**

f. 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. **[5 marks]**

The report should include a title page (including your name and University 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. **[5 marks]**

Viva

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

Marking Scheme

Marking criteria		Marks
A.	**Coding Part**	**57 Marks**
	1. GUI and main method 2. Functionality of Buttons 3. Reading input, checking input and displaying appropriate messages 4. Program Style	12 Marks 20 Marks 13 Marks 12 Marks
B.	**Report Structure and Format**	**43 Marks**
	1. Class Diagram 2. Pseudocode 3. Method Description 4. Test-1(Compiling & Running using command prompt) 5. Test-2 (Adding objects of Tutor and Lecturer, grade assignments amount from lecturer, setting the salary and removing the tutor) 6. Test-3(Testing Appropriate Dialog boxes when unsuitable values entered) 7. Error Detection and Correction 8. Conclusion 9. Overall Report Presentation/Formatting	5 Marks 8 Marks 10 Marks 2 Marks 2 Marks 3 Marks 3 Marks 5 Marks 5 Marks
Total		**100 Marks**