LONDON METROPOLITAN UNIVERSITY

FIRST SIT COURSEWORK-2 QUESTION PAPER: **Year Long 2021/2022**

Module Code:	**CS4001NI**
Module Title:	**Programming**
Module Leader:	**Mr. 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 take 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 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 the 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

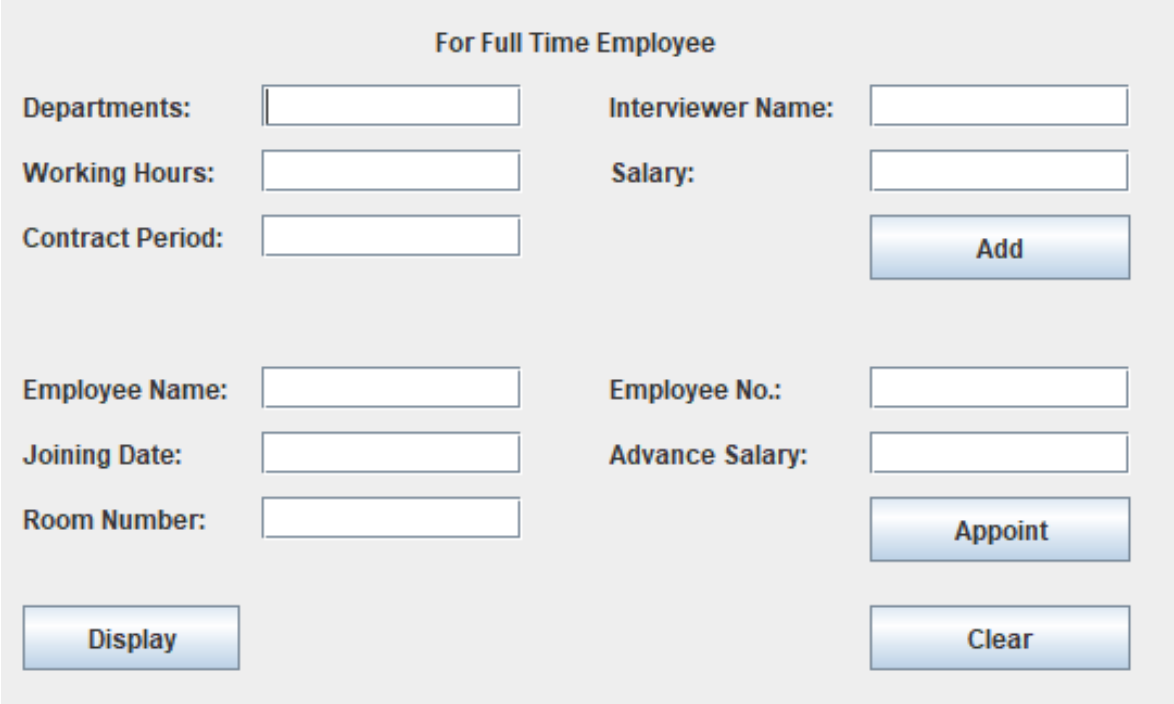
The aim of this assignment is to add a class to the project that you developed for the first part of the coursework to make a graphical user interface (GUI) for a system that stores details of instruments that are sold and rented in an ArrayList. The class will contain the main method and will be tested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

Deliverables

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

Program (60 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 SarangiSansar class should store an array list (not an array) of type Instrument class to hold the InstrumentToRent and InstrumentToSell. There should be **text fields** for entering:

i. Instrument Name
ii. Price
iii. Charge per day
iv. No of days
v. Customer name
vi. Customer phone
vii. Customer PAN No.
viii. Discount Percentage

The attribute **Selling date, Rent date, Return date**, should be entered in a **combo box**.

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

i. **Add Instrument for rent**

When this button is pressed, the input values of the instruments name and charge per day are used to create a new object of type InstrumentToRent which is added to an array list of Instrument class.

ii. **Add instrument for sell**

When this button is pressed, the input values of the instruments name and price are used to create a new object of type InstrumentToSell which is added to an array list of Instrument class.

iii. **Rent the Instrument.**

The Customer Name, Customer phone, Customer PAN No., Rent date, return date, and no of renting days are entered in the GUI. When the rent button is pressed, the input value of the Instrument name is compared to the existing instrument name, and if a valid instrument name has been entered, it is used to rent the appropriate instrument from the list. When the valid Instrument name is entered in the text box, the respective instrument name and charge per day need to be displayed in GUI otherwise display the appropriate message. The method to rent the instrument from the InstrumentToRent class is called here.

Hint: *An object of Instrument is cast as InstrumentToRent*

iv. **Sell the Instrument**

The Customer Name, Customer phone, Customer PAN No.,sell date and discount percentage are entered in the GUI. When the sell button is pressed, the input value of the Instrument name is compared to the existing instrument name, and if a valid instrument name has been entered, it is used to sell the appropriate instrument from the list. When the valid Instrument name is entered in the text box, the respective instrument name and price need to be displayed in GUI otherwise display the appropriate message. The method to sell the instrument from the InstrumentToSell class is called here.

Hint: *An object of Instrument is cast as InstrumentToRent*

v. **Return the Instrument**

The instrument name is entered in the GUI. When the Return button is pressed, the input value of Instrument name is compared to the existing instrument name, and if a valid instrument name has been entered, it is used to return the appropriate instrument from the array list of Instrument. When the valid Instrument name is entered in the text box, the respective instrument name and charges per day need to be displayed in GUI otherwise display the appropriate message. The method to return the instrument of the InstrumentToRent class is called here.

Hint: *An object of Instrument is cast as InstrumentToRent*

vi. **Display**

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

vii. **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 outputs a suitable error message in a message dialog box.

The following parts of the programming will be focused mainly to award the Marks

i. GUI and main method [12 marks]

ii. The functionality of Buttons [26 marks]

iii. Reading input, checking input, and displaying appropriate messages for information as well as error dialog [12 marks]

iv. Programming Style [10 marks]

Report (40 marks)

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

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

b. Pseudocode for each method in each class **[10 marks]**

c. A short description of what each method does **[5 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 a Figure 1 from the command prompt learning aid. **[2 marks]**

Test 2: Evidence should be shown off: **[5 marks]**

a. Add Instrument for rent
b. Add Instrument for sell
c. Rent the instrument
d. Sell the instrument
e. Return the instrument

Test 3: Test that appropriate dialog boxes appear when unsuitable values are entered for the Instrument name, (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 learned 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 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. **[3 marks]**

Viva

Note: *If a 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**	**60 Marks**
	1. GUI and main method 2. The functionality of Buttons 3. Reading input, checking input, and displaying appropriate messages 4. Program Style	12 Marks 26 Marks 12 Marks 10 Marks
B.	**Report Structure and Format**	**40 Marks**
	1. Class Diagram 2. Pseudocode 3. Method Description 4. Test-1(Compiling & Running using command prompt) 5. Test-2 (Adding objects of InstrumentToRent and InstrumetToSell. Renting, Selling and Returning the instrument) 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 10 Marks 5 Marks 2 Marks 5 Marks 3 Marks 3 Marks 4 Marks 3 Marks
	Total	**100 Marks**