LONDON METROPOLITAN UNIVERSITY

FIRST SIT COURSEWORK-1 QUESTION PAPER: **Year Long 2022**

Module Code:	**CS4001NP**
Module Title:	**Programming**
Module Leader:	**Sushil Paudel**

Coursework Type:	**Individual**
Coursework Weight:	This coursework accounts for **30%** of your total module grades.
Submission Date:	**Week 12**
When Coursework is given out:	**Week 8**
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 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 1pm 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.
The viva will be conducted for this assignment.

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

Background

Sarangi Sansar is a musical instruments rental and sales company situated in Kathmandu, Nepal. It is the popular musical instruments store in Nepal providing the retail and wholesale services. It provides the instruments in rent also. It has been using a manual system to keep records of all sales and rental instruments and their customers. Having growth in business of the company, the company feels the need to build a software for handling the daily operation of the company.

Aim

The aim of this assignment is to create a class to represent a musical instrument, together with two subclasses to represent an instrument to rent and an instrument to sell respectively. You will also need to write a report about your program.

Deliverables

Create a new project in BlueJ and create three new classes (**Instrument**, **InstrumentToRent** and **InstrumentToSell**) within the project. **InstrumentToRent** and **InstrumentToSell** are subclasses of **Instrument**. When you are ready to submit your solution, upload your **Instrument.java**, **InstrumentToRent.java** and **InstrumentToSell.java** files (not any other files from the project) together with your report in PDF format.

Program (60 marks)

The program should consist of the following classes (with no additional attributes or methods).

1. The **Instrument** class has five attributes, which correspond to the instrument ID, instrument name, customer name, customer mobile number and Customer Permanent Account Number No. (PAN). Instrument ID is the static numerical variable which will get the automatically incremented values when the new instrument is added in the system. Instrument name, Customer name and customer mobile number represented as string of text. PAN No. represented as number.

The instrument name is initialized in the constructor by being assigned the value of the constructor's parameter. In the constructor, Instrument ID is incremented by 1. Additionally, in the constructor, assign Customer name, customer mobile number as an empty string, PAN is initialized as 0.

Each attribute has a corresponding accessor method. Define setter methods to all the attributes.

A display method will output (suitably annotated) the Instrument ID, Instrument name. If the customer's name, customer mobile no., PAN and no. are not empty or zero, display them also.

[10 marks]

2. The **InstrumentToRent** class is a subclass of the Instrument class and has three attributes:

chargePerDay	- a whole number
dateOfRent	- a string of characters
dateOfReturn	- a string of characters
noOfDays	- a whole number
isRented	- Either true or false

The constructor accepts two parameters, which are instrument name and chargePerDay. A call is made to the superclass constructor with one parameter, instrument name. The chargePerDay is assigned with the corresponding parameters values. Additionally, dateOfRent, dateOfReturn to empty string. noOfDays are set to 0 and isRented is set to false.

Each attribute has a corresponding accessor method. Define setter methods to all the attributes.

There is a method to rent out the Instrument to a customer. The method accepts six parameters, a new customer's name, phone, pan no, the date on which the instrument is rented, the date on which the Instrument is to be returned and no of days to be rented. If the Instrument is not available in the store, an appropriate message will be shown with customer name, phone and return date. If the Instrument is available, the methods to set the customer's name, phone and pan no. are called with respective parameters from the super class. The rented date, the return date and no. of days are all updated by the parameter values input to the method. Set isRented status to true. After successfully renting the instruments, customer details, rented quantity, return date, date of renting, no of days and total charge will be displayed in console. Total charge will be calculated by multiplying the no. of days and charges per day.

There is a method to return the instruments. Inside this method, if the instrument is not in rent, a suitable message is output. If the instrument is in the rent, set the customer's name, phone, pan, return date, rent date to empty string, the number of renting days to 0. isRented status is set to false.

A method to display the details of the instrument is required. It must have the same signature as the display method in the Instruments class. It will call the method in super class to display the instrument id, instrument name. If the instrument has rented already then customer details, renting date and return date should also be displayed. Each output must be suitably annotated.

[20 marks]

3. The **InstrumentToSell** class is also a subclass of the Instrument class and it has three attributes:

price	- a whole number
sellDate	- a string of characters
discountPercent	- a whole number
isSold	- either true or false (Boolean)

The constructor accepts two parameters, which are instrument name and price. A call is made to the superclass constructor with one parameter, instrument name. The price is assigned with the corresponding parameters values. Additionally, sellDate and discount percentage are set to empty string and 0 respectively. And isSold is set to false.

Each attribute has a corresponding accessor method. Define setter methods to all the attributes. While defining the setter method for price, if the instrument has not sold out yet, the new value is assigned to the price attribute. If the instrument is already sold, then a suitable message is output to the user indicating that it is therefore not possible to change the price.

There is a method to sell the instruments. The method accepts, as parameters, customer name, phone, pan no, sellDate and discount percentage. If the instrument is already sold out, an appropriate message including the customer details is displayed. If the Instrument is available, customer details, sell date and discount percentage are required to update by the parameter values input to the method and the isSold status of the Instrument is changed to true. Call the method to set the customer's name, phone, pan no from parents' class with the parameters. Price is calculated by subtracting the discount amount.

A method to display the details of the instrument is required. It must have the same signature as the display method in the Instrument class. It will call the method in Instrument class to display the instrument ID and instrument name. Price of the instrument will be displayed. If the Instrument has sold then customer details, the sold date should also be displayed. Each output must be suitably annotated.

[20 marks]

Additional marks will be awarded for good program style, particularly naming layout and comments.
See http://www.bluej.org/objects-first/styleguide.html for details.

[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: Inspect InstrumentToRent Class, rent the instrument, and re-inspect the InstrumentToRent Class **[2 marks]**
Test 2: Inspect InstrumentToSell Class, sell the instrument, and re-inspect the InstrumentToSell Class. **[2 marks]**
Test 3: Inspect InstrumentToRent Class, return the Instrument, and re-inspect the InstrumentToRent Class **[2 marks]**
Test 4: Display the detail of InstrumentToRent and InstrumentToRent Class. **[4 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

[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]

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**	**60 Marks**
	1. Creating Instrument Class 2. Creating InstrumentToRent Class 3. Creating InstrumentToSell Class 4. Program Style	10 Marks 20 Marks 20 Marks 10 Marks
B.	**Report Structure and Format**	**40 Marks**
	1. Class Diagram 2. Pseudocode 3. Method Description 4. Test-1 5. Test-2 6. Test-3 7. Test-4 8. Error Detection and Correction 9. Conclusion 10. Overall Report Presentation/Formatting	5 Marks 10 Marks 5 Marks 2 Marks 2 Marks 2 Marks 4 Marks 3 Marks 4 Marks 3 Marks
	Total	**100 Marks**