LONDON METROPOLITAN UNIVERSITY

1st Sit Coursework 1 Question Paper: Year Long 2023/2024

Module Code:	**CS4001NI**
Module Title:	**Programming**
Module Leader:	**Mohit Sharma**(Islington College)

Coursework Type:	**Individual**
Coursework Weight:	This coursework accounts for **30%** of your total module grades.
Submission Date:	**12th Week**
When Coursework is given out:	**8th 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 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 implement a real-world problem scenario using the Object-oriented concept of Java that includes creating a class to represent a **Store**, together with its two subclasses to represent a **Department** and a **Retailer** respectively. You will also need to write a report that should contain information about your program.

Deliverables

Create a new project in **BlueJ** and create three new classes (**Store, Department and Retailer**) within the project. **Department** and **Retailer** are **subclasses** of the class **Store**. When you are ready to submit your solution, upload your codes ***Store.java**, **Department.java,** and **Retailer.java*** files (not any other files from the project) together with your report in pdf format.

Program (56 marks)

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

1) The **Store** class has six attributes, which correspond to the **store id, store name, location, opening hour, total sales and total discount**. The **store name, location and opening hour** are each represented as a string of text, **store id** as a whole number and **total sales, total discount** as double.
   The **constructor** accepts four parameters which are **store id, store name, location and opening hour.** The attributes **total sales and total discount** is initialized to zero. Additionally, assign **store id, store name, location and opening hour** with the **parameter values.**

   Each attribute has a **corresponding accessor method**.

   A method is required to **set total Sales**. The method accepts a new **total sales** as a parameter. This method will add the new sales to the previous total sales. And, the new **total sales** will be set to corresponding parameter value.

   A method is required to **set total discount**. The method accepts a new discount as a parameter. This method will add the **new discount price** to **previous discount**. And, the new **total discount** will be set to the corresponding parameter value.

   A display method should output (suitably annotated) the **store id, store name, location, opening hour and total sales.** If the **total sales and total discount** price is **zero**, display a suitable message.

   [10 marks]

2) The **Department** class is also a subclass of **Store** class and it has four attributes:

Product name	- a String
Marked price	- a double
Selling price	- a double
isInSales	- either true or false (boolean)

The **constructor** accepts eight parameters which are **store id, store name,**

location, opening hour, total sales, total discount, product name and marked price. A call is made to the superclass constructor with four parameters and two setter method of **total sales and total discount**. The attribute **isInSales** is set to **true** and **selling price** is set to zero. Also, assign other attributes with corresponding parameter values.

Each attribute has a corresponding **accessor method.**

Create a **mutator method** for attribute: marked price.

There is a method named **calculateDiscountPrice**. This method is used to calculate the selling price of the product. The method accepts isInSales as a parameter and marked price. If the product is available for sales, then, the discount will be calculated based on the marked price:

A → 5000 and above → 20%
B → 3000 and less than 5000 → 10%
C → 1000 and less than 3000 → 5%
D → less than 1000 → 0%
[Note: S.P = MP – MP * discount %]

Now, the attribute selling price will be calculated after the price is discounted. Also, call the mutator method **total discounted price** from the parent class and **calculate selling price** as well. It means the product has been sold and the attribute **isInSales** is set to **false**.

A method to **display** the details of the **Department** is required. It must have the same signature as the display method in the parent class. It will call the method in the super class to display **all the attribute of super class**. If the product is in sales, display **product name** and **marked price** however, if the product is not in sale, display **product name** and **selling price**.

[16 marks]

3) The **Retailer** class is a **subclass** of Store class and has five attributes:

VAT inclusive price	- an integer
Loyalty point	- an Integer
isPaymentOnline	- a Boolean
purchasedYear	- a String

The constructor accepts nine parameters which are **store id, store name, location, opening hour, total sales, total discount, VAT inclusive price, isPaymentOnline and purchasedYear.** A call is made to the **superclass constructor** with four parameters and setter method **total sales** and **total discount**. Additionally, in the constructor, assign **VAT inclusive price, isPaymentOnline, purchasedYear** with the corresponding **parameter value** and **loyalty point** is set to zero.

Each attribute has a corresponding **accessor method.**

Create a **mutator method for attribute: isPaymentOnline**

A method is required to **set** the **loyalty point** as the payment is made from the **online**. The method accepts **new isPaymentOnline** and **new VAT inclusive price** as a parameter and if the payment **is online** then the customer will get loyalty point. The loyalty point will be generated based on the VAT payment by customer, the loyalty point will be 1% of the VAT price. Then the loyalty point will be assigned to the corresponding parameter value.
Note: Loyalty point = 1% of VAT inclusive price

There is a method named **removeProduct.** This method will remove the product if the product doesn't have **loyalty point** and **purchased year** is equal to 2020 or 2021 or 2022, then the product will be removed since a product cannot be older than 1 year. Then, **VAT inclusive price, loyalty point** is set to **zero**, **isPaymentOnline** is set to **false**.

A method to **display** the details of the **Retailer** class is required. It must have the same signature as the display method in the **Store** class. If the product has not yet been removed, it will call the method in the super class to display **all the attribute of super class**. Also, display attributes VAT inclusive price, loyalty point and purchased year. If the product has been removed, call the display details of super class and print a suitable message. Each output must be suitably annotated.

[18 marks]

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

[12 marks]

Report (44 marks)

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

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

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

c. A short description of what each method does **[5 marks]**

d. You should give evidence (through inspection tables and appropriate screenshots) of the following testing that you carried out on your program:
Test 1: Inspect Department class, calculate discount price and reinspect the Department class **[3 marks]**
Test 2: Inspect Retailer class, set loyalty point and reinspect the Retailer class **[4 marks]**
Test 3: Inspect Retailer class again after removing the product.
[2 marks]
Test 4: Display the details of Department and Retailer classes.
[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, semantic or logical errors) should be distinctive and not of the same type. **[3 marks]**

f. The report should contain a conclusion, where you need to include the following things:
   - Evaluation of your work,
   - Reflection on what you learned from the assignment,
   - What difficulties do you encounter 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), an introduction part that contains a brief about your work, 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.

[4 marks]

Viva

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

Marking Scheme

Marking criteria		Marks
A.	**Coding Part**	**56 Marks**
	1. Creating Store Class 2. Creating Department Class 3. Creating Retailer Class 4. Program Style	10 Marks 16 Marks 18 Marks 12 Marks
B.	**Report Structure and Format**	**44 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 3 Marks 4 Marks 2 Marks 4 Marks 3 Marks 4 Marks 4 Marks
Total		**100 Marks**