LONDON METROPOLITAN UNIVERSITY

1st Sit Coursework 1 Question Paper **Year Long 2022/2023**

Module Code:	**CS4001NI**
Module Title:	**Programming**
Module Leader:	**Mr. Ujjwal Subedi** (Islington College)

Coursework Type:	**Individual**
Coursework Weight:	This coursework accounts for **30%** of your total module grades.
Submission Date:	**12th Week (Friday, 27 January 2023)**
When Coursework is given out:	**8th Week**
Submission Instructions:	Submit the following to Islington College's **MST Assignment** Portal before the due date: ● **A report in PDF format and a 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 **Bank Card**, together with its two subclasses to represent a **Debit Card** and a **Credit Card** 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 (**Bank Card**, **Debit Card,** and **Credit Card**) within the project. **Debit Card** and **Credit Card** are **subclasses** of the class **Card**. When you are ready to submit your solution, upload your codes ***BankCard.java****,* ***DebitCard.java,*** *and* ***CrediCard.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 **BankCard** class has five attributes, which correspond to the **card Id, client name, issuer bank, bank account, and BalanceAmount.** The **client name, issuer bank, bank account are** each represented as a **string of text** and **Card ID, and balance amount** as **a number.**

   The **constructor** accepts four parameters which are**, balance amount, card Id,bank account, issuer bank.** The attributes **client name** is initialized to an emptystring. Additionally, **assign balance amount, card Id, bank account, issuer bank** with the **parameter values.**

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

   A **new methods** are required to **set the client name and balance amount**. The method: setclientname accepts a new **client name** as a parameter, whereas, the method:setbalance amount accepts a new **balance amount.** The parameter values is then assigned to the attributes respectively.

A display method should output (suitably annotated) the **card Id, client name, issuer bank, bank account, and BalanceAmount**. If the **client name** is **not assigned, display a suitable message.** **[10 marks]**

2) The **Debit Card** class is also a subclass of **Bank Card** class and it has four attributes:

PIN number	- an integer
Withdrawal Amount	- an integer
date Of Withdrawal	- a String
hasWithdrawn	- either true or false (boolean)

The **constructor** accepts **six** parameters which are **balance amount, card Id,bank account, issuer bank, client name, PIN number.** A call is made to the superclass constructor with four parameters and a setters method:setclientname. Also, assign,**pin number** is assigned with the **corresponding parameter values**.

In the constructor assign the **attribute: hasWithdrawn** to **false**.

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

Create a mutator **method** for attribute: withdrawal amount.

There is a method named **Withdraw.** The method is used to deduct the money directly from the client account. The method accepts **Withdrawal Amount, date Of Withdrawal, and PIN number.** If a valid **pin number** is entered, and **sufficient amount** is also present, then only the amount can be withdrawn and the attribute **hasWithdrawn** is set to **true.** The **attribute: BalanceAmount** is calculated after the withdrawal process.

If the client has entered wrong pin number or there is no sufficient balance, then a suitable message is output to the user.

A method to **display** the details of the **Debit Card** is required. It must have the same signature as the display method in the **Bank Card** class. It will call the method in the **Bank Card** class to display the **card Id, client name, issuer bank, bank account, and BalanceAmount**. It should also display a pin number, withdrawal amount, date of withdrawal. If the **transaction has not** been carried yet, **display the balance amount only**. Each output must be suitably annotated.

[16 marks]

3) The **Credit Card** class is a **subclass** of **Bank Card** class and has six attributes:

CVC number	- an integer
Credit Limit	- a double
Interest Rate	- a double
Expiration Date	- a String of characters
Grace Period	- an integer
isGranted	- either true or false (boolean)

The constructor accepts eight parameters which are **card Id, client name, issuerbank, bank account, BalanceAmount, CVC number, Interest rate, ExpirationDate.** A call is made to the **superclass constructor** with four parameters and asetter method. Additionally, in the constructor, assign CVC number, Interest rate,Expiration Date with the **parameter values**. The attribute:, isGranted is set to **false.**

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

A method is required to **set the credit limit** as each client can have different credit limits. The method accepts a new credit limit, and new Grace Period as a parameter and, if the credit limit is less than or equal to 2.5 times the balance amount, then only the credit is granted to the client as new values are assigned to each attribute. Likewise, the status of isGranted is set to **true**. If the credit has not been granted to the client, then a suitable message is output to the user indicating that the credit cannot be issued.

There is a method named **cancelCreditCard.** This method will remove the client's credit card ~~(only if the client has paid all the interest rates and credits,within the given period).~~ The attributes **CVC number, credit limit and grace period is set to zero.** The attribute **isGranted** is then set to **false**.

A method to **display** the details of the **Credit Card** class is required. It must have the same signature as the display method in the **Bank Card** class. If **isGranted** is set to **true**, It will call the method in the **Bank Card** class to display the **card Id, client name, issuer bank, bank account, and BalanceAmount.** However, if **isGranted** is set to false, **Credit Limit** and **Grace Period** should **not** be displayed**.** Each output must be suitably annotated.

[18 marks]

4) 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 method in 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 the Debit Card class, withdraw the amount, and re-inspect the Debit Card Class **[3 marks]**
Test 2: Inspect Credit Card class, set the credit limit and reinspect the Credit Card class **[4 marks]**
Test 3: Inspect Credit Card class again after cancelling the credit card. **[2 marks]**
Test 4: Display the details of Debit Card and Credit Card 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 Bank Card Class 2. Creating Debit Card Class 3. Creating Credit Card 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**