Deadline
Submit and present latest by **week 14**.

Title
Mini Bank Account System

Learning Outcome
1. Apply modularization and array in programming.
2. Apply object oriented programming concepts in software development.
3. Write object oriented programs in C++ programming language.

Question 1:
Draw the class diagrams for all of the classes as below.
Design a class named 'Account' to represent a simple bank account. The class should hold the following:-

- Fields/attributes – name, ic number, account number, balance.
- Constructors to create an Account object either with a given starting balance or an unknown starting balance in which it should be set to 0.0.
- Methods:
  o all 'getter' & 'setter' methods;
  o display method to print out the details of an Account object;
  o withdraw method & deposit method – which will subtract or add an amount from the balance and output a string containing the revised balance.

Design a new class named 'CurrentAccount' that derives from 'Account' to represent a current account. The class should hold the following additions/changes:-

- Field/attribute – overdraft limit.
- Constructors to create a CurrentAccount object either with a given overdraft limit or an unknown overdraft limit in which it should be set to 500.0.
- Methods:
  o 'getter' & 'setter' methods for overdraft limit;
  o display method to print out the details of a CurrentAccount object;
  o withdraw method – which will subtract an amount from the balance and output a string containing the revised balance. Your program should allow the withdraw method reduce a balance below zero only if the resulting balance is greater or equal to the overdraft limit. If not, an appropriate message should be output.

Question 2:

Develop a program that includes all the classes and a driver program.

In main method, creates an array of 80 current accounts.

Your program should provide the functions such as:
1. Allow user to create a new current account
2. Withdraw / deposit from a current account
3. Print the details of a current account
4. Print the details of all current accounts

There is no limitation for the interface or menu design. Be innovative and creative!!

What to submit
Submit both hardcopy and softcopy of your program design and source code before the due date. You are advice to write your source code in more readable format where indentation is a MUST. Include a cover page with the subject code and title, your name, ID, and submission date as provided. Attach the **Marking Rubric and agreement form** (as the following page) in the last page of your report. Upload your softcopy (your document file with your design and all source code, .cpp file) to moodle and hardcopy of report to your lecturer by the due date.

NOTE:
- You are allowed to complete this assignment in a group of **Maximum 4 (four) members**.
- The work is worth 15 marks.
- 0% for **late submission** regardless of any reasons.
- 0% for both groups if you are caught **copying/plagiarism.**

School of Engineering, Science & Technology
DOP1254 Fundamentals of Object Oriented Programming
Assignment 2 – Marking Rubric

Criteria	Very Poor	Poor	Proficiency	Expert	Marks
Question 1 Class Diagrams (10 marks)	(0 – 1 mark) Does not have any class diagram	(2-5 marks) Partial completion of the class diagram	(6 – 7 marks) Class diagrams completed with minimal requirements	(8 – 10 marks) Class diagrams completed with proper requirements	
Readability of code (indentation/ naming variable / comments) (10 marks)	(0 – 1 mark) Code is highly unreadable	(2-5 marks) Code is readable but with difficulty	(6-7 marks) Code is readable and with little difficulty	(8-10 marks) Code is wholly and easy to read	
Modularity (use of classes/functions, with parameters and returned value) (20 marks)	(0 – 1 mark) Does not demonstrate any proper use of classes/functions	(2-5 marks) Demonstrate some limited use of classes/functions	(6 – 7 marks) Demonstrates proficiency in use of classes/functions	(8 – 10 marks) Demonstrates mastery in the use of classes/functions	X 2
Error reporting capabilities (10 marks)	(0 – 1 mark) Does not demonstrate any error reporting capabilities	(2-5 marks) Demonstrate some limited use of error reporting capabilities	(6 – 7 marks) Demonstrates proficiency in the use of error reporting capabilities	(8 – 10 marks) Demonstrates mastery in the use of error reporting capabilities	
Use of appropriate programming concepts (decision, repetition, etc) (20 marks)	(0 – 1 mark) Does not demonstrate any logical use of appropriate programming concepts	(2 – 5 marks) Demonstrate some limited use of appropriate programming concepts	(6 – 7 marks) Demonstrates proficiency in the use of appropriate programming concepts	(8 – 10 marks) Demonstrates mastery in the use of appropriate programming concepts	X 2
Program execution and output quality (User friendliness, Error free during runtime) (20 marks)	(0 – 1 mark) Program does not able to compile	(2 – 5 marks) Program executed with runtime error but achieve partial program requirements	(6 – 7 marks) Program executed error free with limitations to achieve minimum program requirements	(8 – 10 marks) Program executed error free with correct output and achieve all program requirements	X 2
Report (10 marks)	(0 – 1 mark) Does not submit any proper report	(2-5 marks) Submit report with partial completion	(6 – 7 marks) Submit report with minimal requirements	(8 – 10 marks) Submit complete report with proper format	
				TOTAL (100 marks)	

Signature:……………………. Signature:……………………….
Name: Name:
ID: ID:
Date: Date:

Signature:……………………. Signature:……………………….
Name: Name:
ID: ID:
Date: Date:

*Note: Assignment will not be accepted if this form is not signed and attached with the report.