UGANDA MARTYRS UNIVERSITY

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

Object Oriented Programming
Take-Home Assignment

Submission Guidelines

- Submit your Java code in a .zip file containing the .java source files.
- Write a **README** explaining how to run the program, and include sample input and output.
- **Submit to:** gkasaazi@umu.ac.ug

Objective:

The goal of this assignment is to test your understanding of Object-Oriented Programming principles, including classes, objects, inheritance, and encapsulation.

Instructions:

You are required to create a basic **Bank Account Management System** using Java. Implement the following classes based on the given specifications.

Class 1: BankAccount

- **Attributes:**
  - accountNumber (String)
  - accountHolderName (String)
  - balance (double)
- **Methods:**
  - A constructor to initialize the account details.
  - deposit(double amount): Method to deposit money into the account.
  - withdraw(double amount): Method to withdraw money from the account (ensure the balance does not fall below zero).
  - displayAccountInfo(): Method to display account details (account number, holder name, and balance).

Class 2: SavingsAccount (inherits from BankAccount)

- **Attributes:**
  - interestRate (double)
- **Methods:**
  - A constructor that initializes the savings account with an interest rate.
  - applyInterest(): Method to calculate and add interest to the account balance.

Class 3: CheckingAccount (inherits from BankAccount)

- **Attributes:**
  - overdraftLimit (double)
- **Methods:**
  - A constructor that initializes the checking account with an overdraft limit.
  - Override withdraw (double amount) method to allow withdrawal even if the balance falls below zero (within the overdraft limit).

Task:

1. Create one **SavingsAccount** and one **CheckingAccount** object.
2. Test the following functionalities:
   - Deposit and withdraw for both types of accounts.
   - Apply interest for the savings account.
   - Withdraw within the overdraft limit for the checking account.
3. Display account details after each operation.

Bonus (Optional, 10 marks):

- Implement a **Transaction History** feature that logs all transactions (deposits/withdrawals) for each account.