Banking Inheritance Assignment (Java)

Objective: The goal of this assignment is to design and implement a simple banking system using object-oriented programming concepts in Java, such as inheritance, classes, methods, and method overriding. You will create a system with different types of bank accounts (e.g., SavingsAccount, CheckingAccount), and each type of account will have its own unique behaviors and properties.

Problem Statement:

You are tasked with creating a banking system where different types of accounts inherit from a base class BankAccount. The types of accounts will include SavingsAccount and CheckingAccount. Each account type will have specific features such as deposit, withdrawal, balance checking, interest application, and fees.

Classes to Implement:

Base Class: BankAccount

Attributes:

accountNumber (String): Unique account number for each account.

balance (double): The current balance of the account.

Methods:

BankAccount(String accountNumber, double balance): Constructor to initialize the account number and balance.

void deposit(double amount): Deposits a given amount into the account.

void withdraw(double amount): Withdraws a given amount from the account, ensuring the balance does not go negative.

double getBalance(): Returns the current balance of the account.

String toString(): Returns a string representation of the account details (e.g., account number and balance).

Derived Class: SavingsAccount

Inherits from BankAccount.

Additional Attributes:

interestRate (double): The annual interest rate for the savings account.

Methods:

SavingsAccount(String accountNumber, double balance, double interestRate): Constructor to initialize the savings account with account number, balance, and interest rate.

void applyInterest(): Applies interest to the balance based on the interest rate.

Derived Class: CheckingAccount

Inherits from BankAccount.

Additional Attributes:

overdraftFee (double): A fee charged when the account goes into overdraft.

Methods:

CheckingAccount(String accountNumber, double balance, double overdraftFee): Constructor to initialize the checking account with account number, balance, and overdraft fee.

void withdraw(double amount): Overrides the withdraw method to apply the overdraft fee if the balance goes below zero.

Derived Class: BusinessAccount

Inherits from BankAccount.

Additional Attributes:

monthlyFee (double): A monthly maintenance fee for business accounts.

Methods:

BusinessAccount(String accountNumber, double balance, double monthlyFee): Constructor to initialize the business account with account number, balance, and monthly fee.

void applyMonthlyFee(): Deducts the monthly fee from the account balance.

Instructions:

Create all classes as described above (BankAccount, SavingsAccount, CheckingAccount, and BusinessAccount).

The methods in derived classes (SavingsAccount, CheckingAccount, BusinessAccount) should override or extend the functionality of the base class BankAccount as necessary.

Implement a driver program that:

Creates instances of SavingsAccount, CheckingAccount, and BusinessAccount.

Tests the deposit, withdraw, applyInterest, applyMonthlyFee, and other methods.

Displays the account details and balances after performing operations.