**Assignment 3**

Problem Description:

(*The Account class*) Design a class named Account that contains:

* A private int data field named id for the account (default 0).
* A private string data filed named first name for customer first name.
* A private string data filed named last name for customer last name.
* A private double data field named balance for the account (default 0).
* A private double data field named annualInterestRate that stores the current interest rate (default 0). Assume all accounts have the same interest rate.
* A private Date data field named dateCreated that stores the date when the account was created.
* A no-arg constructor that creates a default account.
* A constructor that creates an account with the specified id, first name, last name and initial balance.
* The accessor and mutator methods for id, name, balance, and annualInterestRate.
* The accessor method for dateCreated.
* A method named getMonthlyInterestRate() that returns the monthly interest rate.
* A method named withdraw that withdraws a specified amount from the account.
* A method named deposit that deposits a specified amount to the account.
* **You should not be able to instantiate object of type Account.** Create two subclasses for checking and saving accounts. A checking account has an overdraft limit, but a savings account cannot be overdrawn. Savings account has transaction fee applied for each transaction (withdraw / deposit).
* **ToString method should be used to display account info  
  ToString method should display all account info including overdraft limit and transaction fee.**

Draw the UML diagram for the classes. Implement the classes. Write a test program that creates a Checking and Savings Accounts objects with an account ID of 1122 & 1123, a balance of $20,000 & 100,000, and an annual interest rate of 4.5%. Use the withdraw method to withdraw $2,500, use the deposit method to deposit $3,000, and print the balance, the monthly interest, account type and the date when this account was created.

Test edge cases.

Analysis:

(Describe the problem including input and output in your own words.)

Design:

(Draw an UML class diagram for the Account class.)

Coding: (Copy and Paste Source Code here. Format your code using Courier 10pts)

**Submission:**

* Complete and Submit this **Word** file to eConestoga before the class on the due day.
* Compile, Run, and Submit your program to eConestoga named as **<Assignment1\_ STUDENT\_ID>.java** (you must submit the program regardless whether it complete or incomplete, correct or incorrect)
* Submission NOT following the above guidelines will NOT be accepted.