**CSC 3250 Object Oriented Design and Development**

**Homework 3 – Due Friday October 13th at noon.**

For this assignment, you will demonstrate your understanding of inheritance, polymorphism, overloading and overriding. Your classes and code must follow the requirements below.

.toString / .equals in every class, returns all member variable statutes

Default / N-arg in every class

Implement the following classes as described:

class BankAccount Need 8 methods

This class acts as a base class in an inheritance structure. The common data to be captured includes a balance and an account number. This class will implement appropriate constructors, a simple accessor and mutator for account number, a simple accessor for balance and mutator methods for balance called deposit and withdraw

class CheckingAccount default constructor to create check number

This class inherits from BankAccount and supports an additional member variable that holds the check number for the last check processed. This class overloads the deposit method since depositing requires a check number and the check number must be validated that it isn't a number less than the current last check processed. If the check number is valid, the deposit method must also update the last check processed value.

class InterestAccount double interestRate

This class inherits from BankAccount and supports an additional member variable that holds the interest rate percentage. This class supports an additional method that applies the interest to the balance.

class FixedDepositAccount Boolean requirement

This class inherits from InterestAccount and supports an additional Boolean member variable that determines whether the account has met the requirement to apply the interest. The account is required to maintain a minimum balance for a specific amount of time before applying the interest. For simplicity, this class will assume that check is performed outside the class and as a result only a simple accessor and simple mutator is needed for this member variable. This class will override the InterestAccount method that applies the interest to the balance, since you can only apply the interest if the time elapsed is true.

All of the above classes will override the toString and equals methods

Implement a program that does the following: ArrayList holds objects in main

Write code to allow a user to create and use Checking, Interest and Fixed Deposit accounts. Main will create a container that can hold any of the accounts. Do not create a separate class to implement the container. Your program should allow a user to add various types of accounts to the container, make deposits and withdrawals from accounts, change the Boolean value in a FixedDepositAccount, and print information formatted in a user friendly way from all accounts or a specific type of account. A menu driven interface should be used to allow a user to perform the various actions.

Grading Rubric:

5 points documentation – header only

15 points – good structure, readability, efficiency

15 points each class implementation (60 total)

20 points – main and functions of main

**PRINT OUT CODE / 10PT FONT / WHITE BACKGROUND**

**NO INPUT VALIDATION**