**JAVA-2**

**Assignment3 worth 3.5%**

A bank operate savings account as per following condition:

**Savings accounts** have a customer id, balance, and overdraft amount. A customer can withdraw an amount of money from the account, but only if they don't go over the overdraft amount. For example, if the balance is $100 and the overdraft is $200, the customer can withdraw no more than $300. The customer can also deposit an amount of money to the account. Please check the sample run of program to design your classes.

Following is the simple structure of abstract class for the program.

You can add more data member and methods in abstract class to accomplish the requirement of program.

pubic abstract class Accounts

{

private String customerID;

private double balance;

public Accounts(String customerID, double balance)

{

this.customerID = customerID;

this.balance = balance;

}

public abstract void deposit(double amount);

pubic abstract void withdraw(double amount);

OR

public abstract double deposit();

public abstract double withdraw();

OR

public abstract double deposit(double amount);

public abstract double withdraw(double amount);

You can use any format of abstract method to do deposit and withdraw.

}

Saving class is a subclasses of Account class.

Write your logic for saving class.

Write a TestProgram to create object of saving class.

Use scanner class to enter value of customerID, balance, overdraft amount, value to be deposit and value to be withdrawn.

Print total balance after withdraw and deposit for saving account.

Use menu driven structure for saving account as per following.

**Following is the sample run of program:**

Please enter a customerID for the savings account: 101

Please enter a starting balance for the savings account: 100

Please enter the overdraft limit for the savings account: 100

**1. Deposit Amount to Savings Account**

**2. Withdraw Amount from Savings Account**

**3. Exit**

**Choose a menu option for the savings account: 1**

Please enter an amount to deposit to your savings account: 50

Amount of $50.0 deposited to account

Current account balance after deposit is $150.0

1. Deposit Amount to Savings Account

2. Withdraw Amount from Savings Account

3. Exit

**Choose a menu option for the savings account: 2**

Please enter an amount to withdraw from your savings account: 80

Amount of $80.0 successfully withdrawn

Current account balance after withdraw is $70.0

Current overdraft balance is $100.0

1. Deposit Amount to Savings Account

2. Withdraw Amount from Savings Account

3. Exit

**Choose a menu option for the savings account: 2**

Please enter an amount to withdraw from your savings account: 100

Overdraft funds required for transaction.

Amount of $100.0 successfully withdrawn

Current account balance is $0.0

Current overdraft balance is $70.0 and your original overdraft limit was: $100.0

1. Deposit Amount to Savings Account

2. Withdraw Amount from Savings Account

3. Exit

**Choose a menu option for the savings account: 2**

Please enter an amount to withdraw from your savings account: 100

Insufficient funds to complete transaction

Current account balance is $0.0

Current overdraft balance is 70.0

1. Deposit Amount to Savings Account

2. Withdraw Amount from Savings Account

3. Exit

**Choose a menu option for the savings account: 3**

Submission and Marking Scheme:

Submit zip file through drop box.

You must also copy and paste all of your source code from all of your classes into a Word document and also submit this document file separately.

Submit this document **in addition to** your source code zip file. **DO NOT add it inside your zip file** - it must be a separate file. This is used for TurnItIn.

Your submissions will be graded with the following consideration:

Marks will be given for originality of the code.

Program is stored in an appropriately named file

Program has the proper header information including your name and program

Program compiles successfully and without warnings

Program is complete and produces correct results

Program uses appropriate indentation to show logical path flows

Program has proper commenting

Marks will be deducted for poor commenting and/or not following Java naming conventions.