**CISC 3115** **Ari Mermelstein**

# Homework #4:

## Overview

In this assignment, you will be asked to implement a Bank simulation program. In this simulation, you will have classes: `Currency`, `Account`, `CD`, `Checking`, `Savings` and `Customer`.

I will provide headers and descriptions for all of the methods that you will need to implement.

The `Currency` class will have one member variable: the number of cents. A Currency object will be used to keep track of the money in a bank account.

**methods:**

```
public Currency();
```

This will initialize the number of cents to 0.

```
public Currency(int cents)
```

This will initialize the number of cents to the specified value.

```
public int getValue()
```

This will return the number of cents in the account.

```
public Currency add (Currency rhs)
```

```
public Currency subtract (Currency rhs)
```

These will add / subtract two Currency objects together.

```
public String toString()
```

This will return a representation for a Currency object, printing the amount of money **in dollars**.

The `Account` class will have the following member variables: A Currency object that will store the balance of the account. The Account class will be an abstract class that will be subclassed by Checking, Savings, and CD.

## methods:

```
public Account(Currency initialDeposit)
```

This creates an account with an initial deposit of "initAmount"

```
public void withdraw(Currency money);
```

Withdraws "money" cents from the account. This will be an abstract method that will need to be overridden.

```
public void deposit(Currency money);
```

Deposits "money" cents into the account. This will be an abstract method that will need to be overridden.

```
Currency getBalance();
```

returns the balance in the account. This will be an abstract method because it must take into account the interest rate if one exists (oh! overloading in English is fun also! ;) )

Now, The Checking, Savings, and CD classes will be subclasses of Account.

A Savings account has an interest rate that gets applied when the balance is checked (let's assume that the balance represents the amount after an interest period). You can withdraw and deposit money from a savings account.

```
public Savings(Currency initial, double rate)

public void deposit(Currency amount)

public void withdraw (Currency amount)

public String toString()
```

A Checking account doesn't have an interest rate (let's say).

```
public Checking(Currency initialAmount)

public void deposit(Currency amount)

 public void withdraw(Currency amount)

public String toString()
```

A CD doesn't allow deposit or withdrawal. You open the account with an initial deposit and then the money cannot be withdrawn until a certain amount if time does past An exception should be thrown.

```
public CD(Currency initialAmount, double rate)

public void deposit(Currency amount)

public void withdraw(Currency amount)

public String toString()
```

The `Customer` class will have 4 member variables: the first name of the customer, the last name of the customer, how many accounts the customer has actually opened, and an arrayList of Accounts.

## Methods:

```
public Customer(String first, String last);
```

```
public void addAccount(Account account);
```

This method will allow the Customer to open an account.

```
public String getFirstName();
```

```
public String getLastName();
```

Accessor methods for the Customer class.

```
public void deposit(Currency money, String type);
```

```
public void withdraw(Currency money, String type);
```

```
public void balance(Currency money, String type);
```

These methods will allow the Customer to withdraw or deposit money from an account. You can assume by protocol that each customer can only have at most one account of each type. Each method should throw an exception if the account doesn't exist or the operation cannot be performed. The withdraw method should throw an exception if the balance would be below 0. Make sure that only one account per type per customer is enforced.

You should create your own custom exceptions for the different situations.

You will also have to create a main program that does the following:

1. Create an array of Customers called *bank*.
2. Create a menu that gives the user the following options:
    O. Become a new customer of the bank. If this is chosen, ask the user for their name, and register them as a customer in the bank. If they are already a customer, print a message that they already have an account.

B. Allows the user to search for the account balance of a particular account. You should prompt the user for their name and account type.

W. Allows the user to withdraw money from a particular account. You should prompt the user for their name and account type. Complain to the user if their account would be overdrawn.

D. Allows the user to deposit money into a particular account. You should prompt the user for their name and account type.

C. Open an account for a user.

Q. Exit the program

Each of the above choices should be implemented in a method. You may wish to make this into a class also for extra credit!

What to submit:

The .java files for all of the classes and the main program and output.