# Classes Using Classes

- A class that contains class member variables demonstrates a *has-a relationship*
  - The class "has a" class
  - For example, a class with a String member variable demonstrates a has-a relationship

## Demonstration Program: Bank

The bank program has two classes:

- Account class: has a member variable (Customer)
- Customer class: has attributes of a customer

A bank maintains accounts where account holders can deposit money and withdraw money. The account holders are customers with a first and last name and complete address.

Class Design:

Account
variables: balance, Customer cust
methods:
getBalance – returns the current balance
deposit – increases the balance. Requires parameter for amount
withdrawal – decreases the balance. Requires parameter for amount. If balance is less than withdrawal, then balance is left unchanged
toString – returns a string with customer information and current balance

Customer
variables: firstName, lastName, street, city, state, zip
methods:
toString() – returns a string with customer information

**Bank Client Code:**

```
import java.util.Scanner;
 import java.text.NumberFormat;

 public class Bank {

      public static void main(String[] args) {
            Account munozAccount = new Account(250, "Maria", "Munoz", "110
Glades Road", "Mytown", "FL", "33445");
            Scanner input = new Scanner(System.in);
            double data;
```

```
            NumberFormat money = NumberFormat.getCurrencyInstance();

            System.out.println(munozAccount);

            System.out.print("Enter deposit amount: ");
            data = input.nextDouble();
            munozAccount.deposit(data);
            System.out.println("Balance is: " +
money.format(munozAccount.getBalance()));

            System.out.print("Enter withdrawal amount: ");
            data = input.nextDouble();
            munozAccount.withdrawal(data);
            System.out.println("Balance is: " +
money.format(munozAccount.getBalance()));
      }
```

**Account Class Implementation:**

```
/**
 * Account class.
 */

import java.text.NumberFormat;

public class Account {
      private double balance;
      private Customer cust;


      /**
       * constructor
       * pre: none
       * post: An account has been created. Balance and
       * customer data has been initialized with parameters.
       */
      public Account(double bal, String fName, String lName, String str,
String city, String st, String zip) {
            balance = bal;
            cust = new Customer(fName, lName, str, city, st, zip);
      }


      /**
       * Returns the current balance.
       * pre: none
       * post: The account balance has been returned.
       */
      public double getBalance() {
            return(balance);
      }


      /**
       * A deposit is made to the account.
       * pre: none
       * post: The balance has been increased by the amount of the deposit.
       */
      public void deposit(double amt) {
```

```
			balance += amt;
		}


		/**
		 * A withdrawal is made from the account if there is enough money.
		 * pre: none
		 * post: The balance has been decreased by the amount withdrawn.
		 */
		public void withdrawal(double amt) {
			if (amt <= balance) {
				balance -= amt;
			} else {
				System.out.println("Not enough money in account.");
			}
		}


		/**
		 * Returns a String that represents the Account object.
		 * pre: none
		 * post: A string representing the Account object has
		 * been returned.
		 */
		public String toString() {
			String accountString;
			NumberFormat money = NumberFormat.getCurrencyInstance();

			accountString = cust.toString();
			accountString += "Current balance is " + money.format(balance);
			return(accountString);
		}
}
```

**Customer Class Implementation:**

```
/**
 * Customer class.
 */
public class Customer {
	private String firstName, lastName, street, city, state, zip;


	/**
	 * constructor
	 * pre: none
	 * post: A Customer object has been created.
	 * Customer data has been initialized with parameters.
	 */
	public Customer(String fName, String lName, String str, String c,
String s, String z) {
			firstName = fName;
			lastName = lName;
			street = str;
			city = c;
			state = s;
			zip = z;
	}
```

```
        /**
         * Returns a String that represents the Customer object.
         * pre: none
         * post: A string representing the Account object has
         * been returned.
         */
         public String toString() {
               String custString;

               custString = firstName + " " + lastName + "\n";
               custString += street + "\n";
               custString += city + ", " + state + "  " + zip + "\n";
               return(custString);
        }
}
```

**Programming Exercises:**

Modify the Customer class to include changeStreet(), changeCity(), changeState(), and changeZip() methods. Modify the Account class to include a changeAddress() method that has street, city, state, and zip parameters.

Modify the bank application to test the changeAddress() method.

Add your code, including the client code, to the Google Doc: "ICS4U – Activity Submission Form".