CST 338

ATM

1. For this assignment you will develop two classes called ATM and Customer that simulate an imaginary automated teller machine (ATM) on the CSUMB campus. In the assignment, you should also develop a UML class diagram for the ATM class and a jUnit test case.
In the program, we assume that an ATM initially keeps $100.00 cash for customer transactions. Additionally, we assume that there are a total of ten customers combined for the OtterUnion and BOA banks. This is a list of customers with their names, PINs, balances, and banks they have.

NOTE: It may be necessary to overload some methods to take different parameter types (i.e. doubles and ints)

1. Alice, 1234, $5000.00, OtterUnion
2. Tom, 2000, $200.00, OtterUnion
3. Monica, 3000, $50.00, OtterUnion
4. Michael, 7777, $0.00, OtterUnion
5. John, 8000, $500.00, OtterUnion
6. Jane, 2222, $500.00, OtterUnion
7. Robert, 2323, $200.00, BOA
8. Owen, 4455, $50.00, BOA
9. Chris, 8787, $10.00, BOA
10. Rebecca, 8080, $555.55, BOA

Demo Program

The following program presents a demo program that uses the ATM and Customer classes.

```
public class ATMDemo
{
	public static void main(String[] args)
	{
		ATM machine1 = new ATM("OtterUnion");
		ATM machine2 = new ATM(200, "BOA", "Library");
		Customer alice;
		System.out.println("===== Welcome to Demo Program =====");
		System.out.println(machine1);
		System.out.println("");
		System.out.println(machine2);
		System.out.println("\n===== Equality Checking =====");
		System.out.println(machine1.equals(machine2));
		System.out.println("");

		machine1.setATM(100, "BIT");
		machine1.addFund(400);// In this method, we assume that an ATM machine
				      // administrator adds $400 more cash to the machine.
		System.out.println(machine1);
		System.out.println("");
		machine1.displayMenu()

		machine1.withdrawal("Alice", 7777, 10.50); // In the method, we assume
			// that the customer "Alice" wants $10.50 withdrawal with PIN 7777.

		machine1.withdrawal("Robert", 2323, 10.50);
		machine1.withdrawal("Alice", 1234, 10000);
		machine1.withdrawal("Alice", 1234, 10);
		machine1.withdrawal("Alice", 1234, 2000);
```

```
        System.out.println("\n===== Machine Status =====");
        machine1.status();
        System.out.println("");
        if (machine1.isCustomer("Alice")) {
              alice = machine1.getCustomer("Alice");
              System.out.println(alice);
              System.out.println("");
              }
        machine1.deposit("Alice", 1234, 10); // In the method, we assume that
                                             // "Alice" conducts the cash deposit $10
                                             // to the machine with PIN 1234.
        System.out.println("\n===== Machine Status =====");
        machine1.status();
        System.out.println("");

//The following method conducts a money transfer transaction from
// "Alice" to "Tom" about $10.00 dollars.

        if (machine1.transfer("Alice", 1234, 10, "Tom", 2000)) {
            System.out.println("Good transfer!!!\n");
        }
        if (!machine1.transfer("Chris", 8787, 10, "Tom", 2000)) {
            System.out.println("Bad transfer!!!\n");
        }
        System.out.println("\n===== Machine Status =====");
        machine1.status();
        System.out.println("\n===== Thank you! ====="); } }
```

Sample Run of the Demo Program

The following presents a sample result of the demo program.

```
===== Welcome to Demo Program =====
Serial Number: 0
Bank Name: OtterUnion
Location: UNKNOWN
Balance: 100.00

Serial Number: 200
Bank Name: BOA
Location: Library
Balance: 100.00

===== Equality Checking =====
false
Serial Number: 100
Bank Name: OtterUnion
Location: BIT
Balance: 500.00

===== ATM Transaction Menu =====
     1. Withdrawal
     2. Deposit
     3. Transfer
Fail - withdrawal
Fail - withdrawal
Fail - withdrawal
Succeed - withdrawal
```

```
Fail - withdrawal

===== Machine Status =====
     Serial Number: 100
     Bank Name:
     Location: BIT
     Balance: 490.00
     5 Transactions so far:
          Withdrawal: 5 (1 success, 4 fail)
          Deposit: 0 (0 success, 0 fail)
          Transfer: 0 (0 success, 0 fail)

Alice: Balance $4990.00

Succeed - deposit

===== Machine Status =====
     Serial Number: 100
     Bank Name: OtterUnion
     Location: BIT
     Balance: 500.00
     6 Transactions so far:
     Withdrawal: 5 (1 success, 4 fail)
          Deposit: 1 (1 success, 0 fail)
          Transfer: 0 (0 success, 0 fail)

Succeed - transfer
Good transfer!!!
```

```
Fail - transfer
Bad transfer!!!



===== Machine Status =====
     Serial Number: 100
     Bank Name: OtterUnion
     Location: BIT
     Balance: 500.00
     8 Transactions so far:
          Withdrawal: 5 (1 success, 4 fail)
          Deposit: 1 (1 success, 0 fail)
          Transfer: 2 (1 success, 1 fail)
===== Thank you! =====
```

Read the demo program and sample execution result very carefully to identify operations of the classes. And also, you should consider the failure cases of the transactions such as incorrect name, incorrect PIN, not enough fund, unmatched bank, etc.

UML Design

Based on the demo program and sample run, identify instance variables and methods for the ATM class and draw the class in a UML class diagram using the Visio program (Google Drawing, Draw.io, etc.). In the UML diagram, you should include all instance variables and methods that are necessary to run the demo program. But you don't need to include constructors, accessors, and mutators because they are obvious. And also, you don't need to draw the UML diagram for the Customer class because it's very small. You should convert the ATM class diagram to a PDF file and submit it on the iLearn. If you submit a different format, you will get zero credit for the UML portion.

Your program will be graded based on

1. Compilation without error.
2. Correct output
3. Good programming structure.
4. Comments. (Title, Abstract, Author, and Date are mandatory.)
5. Meaningful and related variable names.

How to turn in?

Submit your source programs (ATM.java and Customer.java), UML diagram (ATM.pdf) and junit tests on iLearn.