ECS 140B -- Erlang Homework Assignment Number 2
Due no later than 6:00pm Thursday, June 7, 2018

First, this is not a two-person project. It's small enough that it's a one-person homework assignment, so everyone does this one independently -- no teams.

This assignment is focused on concurrent processing in Erlang. As discussed in class, this assignment requires you to write the software that drives a virtual bank and multiple virtual automated teller machines (ATMs). Consequently, you are to create two Erlang modules: 'bank' and 'atm'.

We'll test your code by sending messages from the console to the bank and ATM processes. These messages are operations to be performed by these concurrent processes. The ATM processes will send messages to the bank process when necessary. Similarly, the bank process will send messages to an ATM when necessary.

The bank keeps track of the user account information. The ATMs do not know anything about the user accounts. Any given ATM keeps track of how much cash is available in that ATM. The bank does not know how much cash is in any ATM.

The annotated transcript below (from class) shows the kinds of messages that the bank and ATM processes should be able to handle. Of course, we'll test your program with different accounts, different amounts, different ATMs, etc.

```
% compile the bank module
1> c(bank).
{ok,bank}

% oops, typed 'atn' instead of 'atm'
2> c(atn).
atn.erl: no such file or directory
error

% compile the ATM module
3> c(atm).
{ok,atm}

% spawn a bank process and two ATM processes
4> B = spawn(fun bank2:loop/0).
<0.72.0>
5> A1 = spawn(fun atm2:loop/0).
<0.74.0>
6> A2 = spawn(fun atm2:loop/0).
<0.76.0>
```

```
% tell the bank process to initalize some accounts
7> B ! {create, [{betty,200},{sally,300}], bank1}.
bank bank1 created
{create,[{betty,200},{sally,300}],bank1} % the process returns
% the message it received to the sending process: the console;
% from this point on, I'll edit most of those out of this transcript

% add an additional account - new accounts can only be opened in the
% bank process - you can't go to an ATM to open a new account
8> B ! {open,dave,100,bank1}.
new account dave opened with 100 dollars

% customers can make other transactions at the bank, but this system is
% really designed with the expectation that customers use the ATMs once
% they've opened their accounts (note also that I use the word
% "designed" with some reservation..."thrown together" is perhaps
% more accurate)
9> B ! {balance, sally, bank1}.
account sally has 300 dollars

% tell the ATM processes to start up with $200 in one ATM and
% $500 in the other
10> A1 ! {start, B, bank1, 200, atm1}.
atm atm1 started with 200 dollars cash available
11> A2 ! {start, B, bank1, 500, atm2}.
atm atm2 started with 500 dollars cash available

% ask this ATM how much cash it has
12> A1 ! {cashsupply,atm1}.
atm atm1 has 200 dollars on hand

% using an ATM, sally deposits $50 to her account -- to do this, the
% ATM will have to send a message to the bank -- the bank process then
% updates the account and prints "account sally ..." on the console
13> A1 ! {deposit,sally,50,atm1}.
account sally now has 350 dollars % this message is printed by the bank

% checking the cash on hand in this ATM -- note that the deposit
% does NOT add to the cash on hand for this ATM
14> A1 ! {cashsupply,atm1}.
atm atm1 has 200 dollars on hand % this is printed by the ATM

% sally withdraws $100 -- the ATM sends a message to the bank, and the
% bank updates the account and displays an appropriate message on the
% console -- then the bank sends a message to the ATM confirming the
% withdrawal so that the atm can dispense the cash
15> A1 ! {withdraw,sally,100,atm1}.
100 dollars withdrawn % printed by the bank
account sally now has 250 dollars % printed by the bank

% looking at the updated amount of cash in this ATM -- the amount was
% reduced by the ATM when the ATM received the confirmation message
% about the withdrawal
16> A1 ! {cashsupply,atm1}.
atm atm1 has 100 dollars on hand % this is printed by the ATM
```

```
% sally verifies her balance at a different ATM
17> A2 ! {balance,sally,atm2}.
account sally has 250 dollars % printed by the bank

% dave checks his balance at ATM A1...
18> A1 ! {balance,dave,atm1}.
account dave has 100 dollars % printed by the bank

% ...then he withdraws some money from ATM A2
19> A2 ! {withdraw,dave,50,atm2}.
50 dollars withdrawn % printed by the bank
account dave now has 50 dollars % printed by the bank

% we check the cash supply at atm A1 to make sure that taking
% money from A2 did not affect the cash supply at A1
20> A1 ! {cashsupply,atm1}.
atm atm1 has 100 dollars on hand % printed by the ATM

% dave tries to withdraw $150 from atm A1, but A1 only has $100 cash
% atm A1 stops the transaction without talking to the bank
21> A1 ! {withdraw,dave,150,atm1}.
sorry, insufficient cash in this atm % printed by the ATM

% see, atm A1 only has $100 cash
22> A1 ! {cashsupply,atm1}.
atm atm1 has 100 dollars on hand % printed by the ATM

% atm A2 has plenty of cash
23> A2 ! {cashsupply,atm2}.
atm atm2 has 450 dollars on hand % printed by the ATM

% dave tries to withdraw $150 without success
% this transaction is stopped by the bank
24> A2 ! {withdraw,dave,150,atm2}.
sorry, account dave has only 50 dollars % printed by the bank

% since the bank stopped the transaction, the atm did not dispense any
% money and still has $450 in its supply
25> A2 ! {cashsupply,atm2}.
atm atm2 has 450 dollars on hand % printed by the ATM

% dave withdraws $25 from atm A2
26> A2 ! {withdraw,dave,25,atm2}.
25 dollars withdrawn
account dave now has 25 dollars % printed by the bank

% atm A2 now has $25 less cash than it did before
27> A2 ! {cashsupply,atm2}.
atm atm2 has 425 dollars on hand % printed by the ATM

% again, the withdrawal at atm A2 has no impact on atm A1
28> A1 ! {cashsupply,atm1}.
atm atm1 has 100 dollars on hand % printed by the ATM
```

```
% this system is brittle ... send an erroneous message to a process
% and the process will die
29> A1 ! {cashsupply,foo}.
{cashsupply,foo}
30>
=ERROR REPORT==== 30-May-2018::11:40:05 ===
Error in process <0.74.0> with exit value:
{badarg,[{ets,lookup,[foo,data],[]},
         {atm2,loop,0,[{file,"atm2.erl"},{line,10}]}]}

% yes, atm A1 is dead...no response to the message
30> A1 ! {cashsupply,atm1}.
{cashsupply,atm1}

% but atm A2 is still up and running
31> A2 ! {cashsupply,atm2}.
atm atm2 has 425 dollars on hand % printed by the ATM
32> A2 ! {balance,dave,atm2}.
account dave has 25 dollars % printed by the bank

% and so on, and so on, ...
33>
```

This assignment is purposely underspecified. How you make it all work is up to you, so long as you stay within the constraints given in this document and in class. We may have to make changes in this assignment as questions arise, so stay flexible.

Note that no transactions should ever result in a bank account or cash supply going negative.

Submit your solution as two files (modules) named "bank.erl" and "atm.erl".