Semaphores:

* cust\_ready\_teller
  + Determines whether a customer is ready (in the teller line).
  + Initial value = 0
* cust\_ready\_loan
  + Determines whether a customer is ready (in the loan line).
  + Initial value = 0
* teller\_ready
  + Determines whether a teller is open.
  + Initial value = 2
* loanofficer\_ready
  + Determines whether a loan officer is open.
  + Initial value = 1
* access\_balance
  + Enforces mutual exclusion of modification of balance.
  + Initial value = 1
* access\_loan\_amt
  + Enforces mutual exclusion of modification of the loan amount.
  + Initial value = 1

Functions:

**class Customer**

Integer balance

Integer thread\_number

Integer num\_times\_visited

void run()

{

times\_visited\_bank[cust\_number]++

if( times\_visited\_bank == 3)

return

signal(cust\_ready\_teller)

wait(teller\_ready)

choice = rand(0 or 1 or 2)

amount = rand(100 or 200 or 300 or 400 or 500)

}

**class BankTeller**

Integer thread\_number

void run()

{

wait(cust\_ready\_teller)

if(choice = 0)

deposit(amount)

if(choice = 1)

withdraw(amount)

signal(teller\_ready)

}

Integer deposit(Integer amount)

{

wait(access\_balance)

balance = balance + amount

signal(access\_balance)

}

Integer withdraw(Integer amount)

{

wait(access\_balance)

balance = balance – amount

signal(access\_balance)

}

**class LoanOfficer**

void run()

{

wait(cust\_ready\_teller)

loan(amount)

signal(loanofficer\_ready)

}

Integer loan(Integer amount)

{

wait(access\_loan\_amt)

loan\_amt = loan\_amt + amount

signal(access\_loan\_amt)

}

**class Main**

{