Fall Term 2015

CS 439 - Principles of Computer Systems
Project 3

Assigned on: **Oct 15 2015**
Due by: **Oct 26 2015**

1 Bank Teller Problem (40 Points)

The following is known as the classic "bank teller problem":

A bank has a variable number of tellers who check in to work by calling `teller_check_in()`*. Customers enter the bank for service by calling* `do_banking()`*. When there is an available teller,* `do_banking()` *returns the teller and the customer gets service. If there is no available teller,* `do_banking()` *blocks until a teller becomes available before returning it. When customers finish getting service, they call* `finish_banking()` *and leave the bank. Tellers may also check out (if they are not currently serving a customer) by calling* `teller_check_out()`*.*

On the course homepage you can find a code skeleton with a framework for creating teller and customer threads. Your task is to complete the creation of threads, implement mutual exclusion for critical sections using phtread_mutex, and implement the correct synchronization using pthread condition variables.

Note that in this program the tellers have to wait until a potential current customer has finished doing banking before they can check out.

Compile your program (with `-pthread` or, depending on your system, `-lpthread`) and test it. A correct program should run forever while incorrect programs might segfault or deadlock after a while.