

CS 2500 Exam 1 HONORS SUPPLEMENT – Fall 2014

Name: _____

Husky email id: _____

- This supplement to Exam 1 is intended for students enrolled in the Honors section of 2500.
- See the instructions on the regular exam.
- You should design helper functions as needed; as usual, they should be designed according to the recipe.

Problem	Points	/out of
1		/ 12
2		/ 12
3		/ 12
Total		/ 36

Good luck!

Problem 1 Wallykazam Bank offers its clients interest-bearing checking accounts. Accounts with a balance of \$500 or more accrue interest at 2% per month while accounts with a balance below \$100 incur a monthly fee of \$10.

12 POINTS

Here is the data definition that Wallykazam uses to keep track of accounts:

```
(define-struct acct (id balance))
;; An Account is a (make-acct String Number)
;; Interpretation: (make-acct s n) denotes an account
;;   with ID s and a current balance of n dollars
;;
;; An LOA (list of accounts) is a one of:
;; - empty
;; - (cons Account LOA)
;; Note: an LOA may not contain multiple accounts with
;;   the same ID
```

Design a function `monthly` that Wallykazam Bank can use to do monthly processing of their list of accounts. The processing involves paying 2% interest or charging the \$10 fee in accordance with the policy described above, depending on the balance in each account.

You may use the following accounts when developing examples:

```
(define a1 (make-acct "001" 100))
(define a2 (make-acct "002" 50))
(define a3 (make-acct "003" 500))
(define a4 (make-acct "004" 1000))
```

[Here is some more space for the previous problem.]

Problem 2 On any given day, numerous Wallykazam clients may request that their account be closed. The bank would like to process these requests in a batch at the end of each business day. Design a function, `purge-accounts`, that takes a list of existing accounts `loa` and a list of the IDs of accounts that need to be closed, and removes all accounts with IDs in the latter list from `loa`.

12 POINTS

Assume the same data definitions for `Account` and `LOA` as in the last problem. You may use examples of accounts from the last problem when developing tests.

[Here is some more space for the previous problem.]

Problem 3 The actual data definition that Wallykazam Bank uses for accounts is more complicated than what we've shown you thus far. We neglected to mention that for each account the bank must also keep a history of transactions, i.e., all withdrawals and deposits made to the account.

12 POINTS

Here is the actual data definition that the bank uses:

```
(define-struct acct (id balance history))
;; An Account is a (make-acct String Number History)
;; Interpretation: (make-acct s n h) denotes an account
;;   with ID s, a current balance of n dollars, and a
;;   transaction history h

(define-struct th (trans hist))
;; A History is one of:
;; - Number ;; interp: beginning balance at account creation
;; - (make-th Transaction History)

;; A Transaction is a Number.
;; Interpretation:
;;   -- a negative number denotes a withdrawal
;;   -- a positive number denotes a deposit
;; Note: a Transaction may not be zero
```

Here are some examples of accounts:

```
(define a1 (make-acct "001" 100 100))
(define a2 (make-acct "002" 50
                     (make-th -10 (make-th -20 80))))
(define a3 (make-acct "003" 500
                     (make-th 100 (make-th -10 410))))
(define a4 (make-acct "004" 1000
                     (make-th 100
                               (make-th -100
                                         (make-th -100 1100)))))
```

If an account holder disputes a withdrawal, the bank may decide to remove the disputed transaction from the account. To do this, the bank needs a function, `remove-disputed-withdrawal`, that removes the withdrawal from the account's transaction history and updates the account balance to reflect a credit of the disputed amount. Please design this function. It should take an account `acc` and the disputed withdrawal amount (a positive number) as inputs, and modify `acc` by removing one disputed withdrawal transaction from the account's history and crediting the account balance with the disputed amount.

You may assume that the disputed amount is guaranteed to appear as a withdrawal in the given account's history. The order of transactions in the account history should not be changed.

You may abbreviate `remove-disputed-withdrawal` to `remove-dw`.

[Here is some more space for the previous problem.]