HW1: DSSL Warmup

Due: Thursday, October 5, at 11:59 PM, via GSC

You must work on your own for this assigment. Future assignments will allow partners, but not this one.

The purpose of this assignment is to get you programming fluently in DSSL, the language that we'll be using for the course.

In `warmup.rkt`[1] I've supplied headers for the functions that you'll need to write, along with some tests.

Part I: Installing DSSL

To complete this homework assignment, you will first need to install the DrRacket programming environment (available from `racket-lang.org`). Then you will need to install the DSSL language within DrRacket.

Once you have DrRacket installed, open it and choose "Package Manager" from the "File" menu. Go to the "Do What I Mean" tab, and paste the package URL, `https://github.com/tov/dssl2.git`, into the text box. Then click the "Install" button and wait for installation to finish. When it's finished, the "Install" button should change to "Update"; then close the window.

Part II: Struct practice

In `warmup.rkt` we define a data type for representing bank accounts as follows:

```
#lang dssl2

# AccountId is Natural

# Account account(AccountId, String, Number)
defstruct account(id, owner, balance)
# where
```

[1] `http://goo.gl/YBv9gS`

```
#   `id` is the account number,
#   `owner` is the name of the account holder, and
#   `balance` is the balance.

# Examples:
let ACCOUNT0 = account(0, "Alan Turing",    16384)
let ACCOUNT1 = account(1, "Grace Hopper",   32768)
let ACCOUNT2 = account(2, "Ada Lovelace",   32)
let ACCOUNT3 = account(3, "David Parnas",   2048)
let ACCOUNT4 = account(4, "Barbara Liskov", 8192)
```

Write these two functions:

1. `account_credit! : Number Account -> Void` modifies the `Account` by adding the `Number` to its balance.
2. `account_transfer! : Number Account Account -> Void` modifies both `Account`s by transferring the given amount from the first `Account` to the second. That is, its subtracts the `Number` from the first `Account`'s balance and adds it to the second `Account`'s balance.

Part III: Vector practice

Write these four functions:

3. `vector_swap! : VectorOf<X> Natural Natural -> Void` takes a vector and two indices, and swaps the vector's values at the indices.
4. `vector_copy : VectorOf<X> -> VectorOf<X>` copies a vector. That is, it makes a new vector of the same length containing the same elements. (Use a vector comprehension for the easiest way to do this and the next one.)
5. `vector_copy_resize : Natural VectorOf<X> -> VectorOf<X>` copies and resizes a vector. In particular, it takes as its first parameter the length for the new vector to create and then copies over as many elements as will fit. If the new vector is shorter than the old then it won't contain all of the old vector's elements. If the new vector is longer then the remaining elements should be filled with `False`.

6. `find_largest_account : VectorOf<Account> -> Account` takes a non-empty vector of `Account`s and returns the account with the largest balance. You may assume that the vector has at least one element, and you don't need to worry about ties for the largest balance.

Deliverable

The provided file `warmup.rkt`, containing definitions of the six functions described above and sufficient tests to be confident of your code's correctness. You will be graded for correctness, efficiency, style, and adequacy of tests.

Submission

Your homework must be submitted via the online system GSC, which can be found at `https://eecs214.cs.northwestern.edu/`.

Before you can submit, you will have to sign up for an account. **Your user name must be your NetID,** which is three letters followed by three or four digits. If you use anything else as your username, we will not know who gets credit for your work. Please choose a secure password and do not share it.

When logging into GSC, you will usually want to check the "Keeps login for 2 weeks" box, as otherwise refreshing the page will lose your session.

Upon signing into GSC, you should see a list of assignment with due date. Select the assignment you want to submit and upload the file(s). You do not need to do anything to indicate that your submission is complete—whatever you have uploaded as of the due date will be considered your submission.