CS 340, Fall 2016 — Nov 22nd — Midterm 2

First things first: the following web page specifies which resources you may use during the exam, and links to the permitted resources:

  https://ycpcs.github.io/cs340-fall2016/assign/exam02.html

Start by downloading the exam zipfile using the command

  `wget` *zipfileURL*

where *zipfileURL* is the URL of the exam zipfile.

Import the zipfile as an Eclipse project. The project will be called `cs340-exam02`. You will be editing the file `src/cs340_exam02/core.clj`.

There are five functions to complete: **juggle**, **generate-multiples**, **make-multi-applicator**, **count-matching**, and **apply-fn-chain**. There is also one bonus function you can implement for extra credit: **circus**. Each function has a detailed comment explaining the requirements, expected behavior, and a point value.

Make sure you meet all of the requirements of each function: e.g., if the function is required to be tail recursive, make sure your function is tail recursive.

Some general hints:

- `(empty? s)` tests whether sequence `s` is empty
- `(first s)` gets the first element in sequence `s`
- `(second s)` gets the second element in sequence `s`
- `(rest s)` gets a sequence with all but the first element of sequence `s`
- `[]` is an empty vector
- `(conj v elt)` creates a vector which results from appending `elt` onto the vector `v`
- `(conj v elt1 elt2)` returns a vector which results from appending `elt1` and `elt2` onto the vector `v`
- `(concat s1 s2)` returns a sequence containing all of the elements of sequence `s1` followed by all of the elements of sequence `s2`

You can run the command `lein test` in a terminal window to run unit tests. Note that as you are working on a function, testing it interactively in a REPL (using the example inputs described in the function comment) will probably be the most useful form of testing.

When you are done, run the command `make submit` in a terminal. Type your Marmoset username and password when prompted.

Most importantly:

Have fun!