CIS 130 - Week 4 Lab Exercise
DUE: End of lab, Wednesday, February 10th

Purpose: to practice using the design recipe, writing and using named definitions, writing `modulo` expressions, and composing these together to create a specified animation

You will be assigned to **pairs** to work on this lab exercise (and, yes, working in pairs is **required**): 2 people sharing 1 computer, one typing (driving) while the other says what to type (navigating). Switch roles about every half-hour. (If we have an odd number of students, there will be one trio.)

Remember: use of the Design Recipe is now required for all functions that you write.

You will receive full credit for this lab exercise IF:

* you submit the results from problems below by about 5 minutes before the end of lab (or, for significant partial credit, whatever you have completed by that point), using **~st10/130submit**, as homework number **44**, and

* you meet the specifications below, including putting in all of the required design recipe elements for the scene function (contract, purpose, check-expects, and header/body). For example, **having a function body but no check-expects for that function WILL cause you to lose points on this lab exercise!!** And on every other function throughout the semester... 8-))

1. Open DrScheme, and put a comment including **both** of your names, and that this is the **Week 4 Lab**.

2. Find an image on the web that you would like to move around in a scene, and give it a name within DrScheme. (That is, write a `define` expression giving a name to this image.)

3. Write `define` expressions giving the values you want for a desired scene `WIDTH` and `HEIGHT`. (That is, make `WIDTH` and `HEIGHT` named constants, whose values are the width and height you want for the scenes you are about to create.) For the rest of this exercise, use `WIDTH` and `HEIGHT` to represent the scene's width and height in expressions you write.

4. Remember how we used `place-image` and `empty-scene` to create a "backdrop" scene for an image to move in in class? Now you should define a name `BACKDROP` that has as its value a desired scene with at least three visible images "placed" within it. (You can put whatever you'd like -- circles, rectangles, etc. I've posted a handout, "Elaborating DrScheme scenes", with descriptions of a few more functions that create images on the public course web page along with this handout.)

After this definition, put the now-simple-expression `BACKDROP` in your Definitions window, so your backdrop will appear in the Interactions window when the Definitions window contents are Run.

5. Now, remember the `modulo` function we discussed in class? It returns the integer remainder from division -- it has the lovely behavior that if you use modulo on any value and a given

value, the result will ALWAYS be between 0 and one less than that given value. (That is, `(modulo anything 100)` will always be between 0 and 99, no matter what the value of `anything` is, because the integer remainder from dividing anything by 100 has to be between 0 and 99...)

Write at least three different `check-expect` expressions in which the first expression is a `modulo` expression, and the second is the value that `modulo` expression should have, to help make sure you understand this very-useful function.

6. Now, design a function `create-lab-scene` that expects a time-counter value, and produces a scene that:
* places your image from problem 2 in your backdrop from problem 4,
* ...such that the time-counter helps to determine at least the x-coordinate where that image will be placed in the scene, (you can have it affect the y-coordinate, **also**, IF you want to -- your choice);
* ...using `modulo` so that, no matter what the time counter is, the image will always be placed *within* the scene (at least part of your image will always be visible in the scene, no matter how big the time counter is)

In writing your check-expects for this function, include at least 2 check-expects, and make sure that at least one of your check-expects is for a time-counter value larger than both the `WIDTH` and `HEIGHT` of your backdrop.

7. Finally, write an `animate` expression to use your `create-lab-scene` function to create an animation.

Remember: however far you get by about 5 minutes before the end of lab, **submit** your definitions file using **~st10/130submit** and a homework number of **44**. Make sure that BOTH (or all three) students save a copy of the definitions file for themselves, as well!