

COMPUTER SCIENCE

LAB 2

WORKING with FUNCTIONS and IMAGES

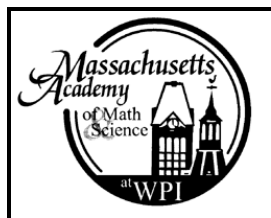
Adapted from exercises written by Kathi Fisler, Worcester Polytechnic University and Stephen Bloch, Adelphi University

For the following exercises, you will need to download the image teachpack and add it (under Language menu in Scheme). See the helpdesk in Dr. Scheme to look at the functions available in the teachpack.

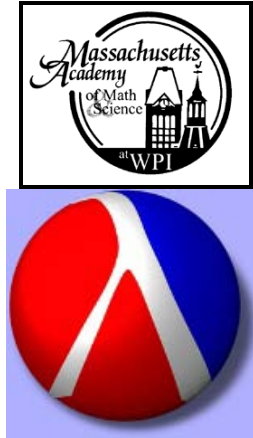
1. Using the functions defined in *image.ss*
 - o Write an expression (or set of expressions) that draws a sail boat with two sails, floating on water in a 200 x 200 pixel box. The box should be visible using a solid black outline.
 - o Write an expression (or set of expressions) that draws a car driving on a street in a 200 x 200 box. The box should be visible using a solid black outline.

For the following exercises, choose 3-5 small (< 300x300 pixels) images from the internet (or copy and paste the images below) for your examples and test cases. Insert the images and define them as constants before any of your functions.

2. Write a function, *side-by-side-images*, that consumes two images and returns a new image with the first image immediately to the left of the second image. You do not need to center the images vertically, but you may if you wish. Make sure to test your function fully with different sized images to make sure it works in all cases.



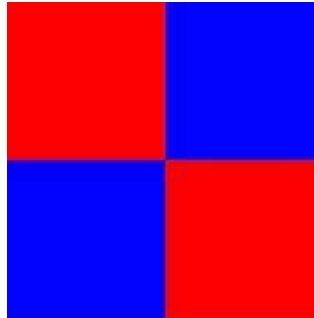
3. Write a function ***stack-images*** that consumes two images and returns an image with the first image just above the second image, as shown below. You do not have to center the images horizontally, but you may if you wish.



4. Develop a function named ***counterchange*** which takes in two images and returns a 2x2 arrangement of them, as shown below. Be aware that the resulting images should be left justified (no shift on bottom image, and no overlaps). Consider using previously written functions to complete this task!



5. Develop a function named **checkerboard** which takes in two colors (e.g. symbols) and returns a 2x2 arrangement of them, as shown below. Make each square of the checkerboard 50x50 pixels.



DESSERT



6. Develop a function **8x8-checkerboard**, which takes two colors and creates a checkerboard with 8 squares across and down.
7. Design and write a function **three-side-by-side-images** that takes three images for input and outputs an image of the three side by side. They should not overlap and should abut each other exactly.
8. Using the built-in Scheme **random** function (use helpdesk to see how it works), develop a function called **random-star** that consumes a color (symbol) and produces a 200x200 box with a randomly placed star in the color specified within the box. The star should have an inner radius of 5 and an outer radius of 30 (see *image.ss* documentation for **star** function). The star should fit completely in the box.
9. Design a function, **inscribe-shapes**, which takes a number as input. The number represents the radius of a circle. The function will output a red circle of the given radius. Overlaid on top of the circle is a blue square inscribed in the circle. Overlaid on top of the blue rectangle is a yellow equilateral triangle, inscribed in the circle too. The output should look like the figure below.

