

# CS 2500, Lab 1

---

## Online Submission Setup

All labs will be turned in jointly via the automated homework server. Download the plugin for DrRacket and set up the handin client following the instructions on [this page](#).

---

## Pair Programming [1 min]

You will be working in pairs during labs. Pairs consist of a pilot and a co-pilot. The pilot types at the keyboard while the co-pilot looks over the pilot's shoulder. Remember if the plane goes down, you are both in trouble.

### Choose a Partner!

---

## Finger Exercises

Start DrRacket. Make sure that the language level is set to “Beginning Student” (*Note:* if you change the language level, it will not take effect until you hit the “Run” button).

*Exercise 1:* Experiment using DrRacket's *Interactions Window* as a calculator. See what operations you can use to calculate. Make sure to try big numbers and fractions. [3 min]

*Exercise 2:* Design a function that when given an integer that represents distance in miles will return the time (in hours) it takes to travel that distance when going 60 MPH. Write your function in the *Definition Window*, click **Run**, then use the *Interactions Window* to test your function. [5 mins]

*Exercise 3:* Design a function that given an integer representing time (in minutes) will return the distance traveled when going 70 MPH, rounding up to *whole* miles. [15 mins]

*Hint:* DrRacket comes with a help system called Help Desk. Open the Help Desk and click “How to Design Programs Languages,” then click “Beginning Student.” You will get a reference for the *Beginning Student language*. You can also highlight the name of a function and press F1, this will search Help Desk for the highlighted text.

*Exercise 4:* There are three kinds of errors you can make when writing a program in DrRacket. Come up with examples of each and try them out. Examine any error messages you get back. [5 mins]

1. syntax error

2. run-time error
3. logical error

---

## Partner Switch [1 min]

Pilots stand up and switch with your co-pilots.

---

## Images and Scenes [50 mins]

Access the 2nd edition HTDP “universe” and “image” libraries in DrRacket. Put the following magic text at the top of your definitions window:

```
(require 2htdp/image)
(require 2htdp/universe)
```

Libraries come with documentation. In the Help Desk, search for “universe.” Also have a look at the documentation of the image library titled “2htdp/image.”

*Exercise 5:* Download an image off the web and insert it into your Definition window. Use **define** to give it a descriptive name. Examples: cars, boats, planes etc.

*Exercise 6:* Use **place-image** and **empty-scene** to create scenes with your image.

*Exercise 7:* Create rectangles with the following dimensions:

1. 10x25
2. 20x50
3. 30x75
4. 40x100
5. 50x125
6. 60x150

*Exercise 8:* Design the function **rec-sequence** that given an index will return the corresponding rectangle, e.g., (**rec-sequence** 2) returns a rectangle of size 20 by 50.

## Animation

Using the “universe” module we can create animations. The module allows you to model a *world*. Every time the clock ticks, the module uses one of your functions to update or create a new world, which becomes the current world. The module can use another of your functions to create an image of your world.

1. Our “world” will represent the number of ticks passed,

```
;; World is a positive number (current time)
```

2. Create a function that will calculate the *next* world

```
;; World -> World
;; calculates the next world (increments the time)
(define (next-world w)
  (add1 w))
```

3. Create a function that will create an image snapshot from a world

```
;; World -> Image
;; draw the image for snapshot time w in the middle
;; of a 200x200 scene
(define (world-draw w)
  (place-image (circle w ‘‘solid’’ ‘‘black’’)
    100 100
    (empty-scene 200 200)))
```

---

## Switch Partners

---

4. Genesis (lookup `big-bang` in Help Desk).

```
;; We start at time 1, use next-world to update the world at each
;; tick, and use world-draw to render the world as a scene:
(big-bang 1
  (on-tick next-world)
  (on-draw world-draw))
```

*Exercise 9:* Change the animation above so that instead of growing larger, the disk moves across the canvas. Try left to right ( $x$ ), top to bottom ( $y$ ) and diagonal (both  $x$  and  $y$ ).

*Exercise 10:* Read the documentation of the “image” module and create an image of a car using circles and rectangles.

*Exercise 11:* Use the car image from Exercise 10 and create an animation with the car moving across the canvas.

---