

## Lab: Records

- Exercises
  - Exercise 0: Preparation
  - Exercise 1: Testing Compounds
  - Exercise 2: Filtering Compounds
  - Exercise 3: Falsifying Compounds
  - Exercise 4: A Shirt Record
  - Exercise 5: Sorting Shirts
  - Exercise 6: Searching Shirts
- Notes

### Exercises

#### Exercise 0: Preparation

- Make a copy of `compound.ss`.
- Scan through that file to make sure that you understand all the procedures.
- Start DrScheme.

#### Exercise 1: Testing Compounds

- Try the compound examples given in the reading on records to see if they behave as I said they would.
- Try a few of your own variations.
- Explain why the attempted fake is not listed as a compound. Hint: Think about the order of the `let` and `lambda` in `produce-type-mark`.

#### Exercise 2: Filtering Compounds

Define a procedure `filter-by-color` that takes two arguments -- a list `ls` of records of type `compound` and a color symbol -- and returns a list of the names of compounds of that color that are elements of `ls`.

#### Exercise 3: Falsifying Compounds

- Can you set the melting point of a compound to an illegal temperature? If so, how? If not, why not? (Note that there are other ways to modify compounds than to use the record's provided procedures.)

b. Find a way to make a vector that `compound?` identifies as a `compound`. You may not use `make-compound` or `compound-copy` to build your new vector. However, you may use either procedure to create a “helper” `compound` from which you copy values.

### **Exercise 4: A Shirt Record**

Write a set of definitions for a record type `shirt`, to be used in a program that keeps track of the inventory of a clothing store. Provide fields for catalog number, intended gender of wearer, size, color, price, and quantity in stock. Only the last two fields should be mutable.

### **Exercise 5: Sorting Shirts**

Define a procedure that sorts a vector of records of type `shirt` into ascending order by catalog number.

### **Exercise 6: Searching Shirts**

Adapt the binary search procedure so that it takes two arguments -- a vector `vec` of records of type `shirt`, sorted by the procedure defined in the previous exercise, and a catalog number `catno` -- and returns the entire record that contains that catalog number, if there is one in `vec`, or `#f` if there is no such record.

## **Notes**