

## Lecture 7: Records and List Comprehensions

**Exercise 7.1** Based on the definition of `ABCD`, what are the types and behaviors of the functions `foo`, `bar`, and `baz`? Think about it, and then test them out.

**Exercise 7.2** Should the following definition be acceptable? Think about it, and then try it out.

```
data Data = One { data :: Int } | Two { data :: Bool }
```

**\*Exercise 7.3** The file `Records.hs` defines two `Teachers`, `professorChugh` and `professorKurtz`, and a "database" of `Students`, `allStudents`.

Implement the function

```
studentsOfTeacher_ :: [Person] -> Person -> [((Int, Int), [(String, String)])]  
studentsOfTeacher_ students teacher = undefined
```

to return those `students`, identified by lastname-firstname pairs, enrolled in each of the `teacher`'s courses. Your implementation can assume that `students` is a list of `Student` values and that `teacher` is a `Teacher` value.

For example:

```
> studentsOfTeacher professorChugh  
[((16100,1),[("Student","B"),("Student","STEAM")])]  
  
> studentsOfTeacher professorKurtz  
[((16100,2),[("Student","C")]),((28000,1),[("Student","D"),("Student","E")])]
```

Once you are done, consider how one might eliminate the assumption above (but don't submit this).