COSI135: Fall 2014
Assignment #2

Before you begin, it is strongly recommended that you review chapter 8 of Learn You A Haskell: http://learnyouahaskell.com/making-our-own-types-and-typeclasses. You should load and read the contents of `dataTypeDemo.hs` and `classDemo.hs`. You should also be familiar with the correspondence between representing lambda formulas in Haskell (e.g. `(\ x -> run x) spot`) and in conventional notation (e.g. λx.run(x)@spot).

In GHCI (or your Haskell interpreter of choice), Load `catch.hs` and run the following command: `(\ f x y -> f x y) catch "Mary" "the bus"`

1. In `catch.hs` you are given a custom datatype Object that contains a name field. Modify the "catch" function so that it takes arguments of type Object instead of type String. You will need to create Objects that the new function can accept in this file (HINT: the command you'll give the interpreter for the new function should look like "`(\ f x y -> f x y) catch boy ball`" where <`boy`> and <`ball`> are of type Object).

2. Next, create a new file "multiCatch.hs," that will allow "catch" to take arguments of multiple types (you will need to create these types – suggested types: Human, Animal, Vehicle – all types only need to contain a field for a name; that is, they can be structured identically to Object above). Use classes and instances to create code for a "catch" predicate that makes sense for the semantics of the verb, i.e. a human can catch a bus (vehicle), and a cat (animal) can catch a mouse (animal), but other combinations may not be valid. There should be at least two formulations of this new "catch." We should be able to pass your code "`(\ f x y -> f x y) catch a b`" where <`a`> and <`b`> are instances of your new datatypes.

3. Once you've accomplished this, do the same for at least one more transitive verb whose sense changes depending on the semantics of its arguments (for example, "play", as in "Babe Ruth plays baseball" vs. "Yo-Yo Ma plays the cello", or "serve", as in "Elected officials serve the people" vs. "Olive Garden serves food" — we are obviously not too concerned with the truth of these statements!). What datatypes do you need to define to make sense of the difference senses of the verb? As with problem 2, you should be able to allow and disallow certain combinations of arguments for the verb you choose to implement. Please put the code for this verb in a new .hs file, entitled <verb>.hs. In comments or a separate file, please explain what datatypes your chosen verb can take and what restrictions you've placed on them in your code.