```haskell
-- Name: Xioameng Cao
-- Email: xcao07@syr.edu
-- Section: CST 1-109

data Animal = Snake | Tiger | Mongoose | Wasp | Fish
              deriving(Eq, Ord, Enum, Show)

mammal :: Animal -> Bool
mammal Tiger = True
mammal Mongoose = True
mammal _ = False

data Attribute = Brawn | Speed | Wits | GoodLuck
                 deriving (Eq, Ord, Show)

data Creature = Cr String Animal Attribute
                deriving (Eq, Show)

fred, rex, bella :: Creature
fred = Cr "Frederick" Fish Brawn
rex = Cr "Rex the Renegade" Mongoose GoodLuck
bella = Cr "Isabel" Tiger Speed

battle :: Creature -> Creature -> String
battle (Cr name1 sp1 att1) (Cr name2 sp2 att2)
    | att1 > att2 = name1 ++ " wins!"
    | att2 > att1 = name2 ++ " wins!"
    | otherwise = "It's a draw!"

upgrade :: Creature -> Attribute -> Creature
upgrade (Cr name sp att) new
    | new > att = Cr name sp new
    | otherwise = Cr name sp att

upgradeAll :: Attribute -> [Creature] -> [Creature]
upgradeAll new (c:crList) = upgrade c new : upgradeAll
new crList
upgradeAll new [] = []
```