### Project 2. Higher-order functions (part 1)

*Higher-order functions are the heart and soul of functional programming.*

[**LYH**](http://learnyouahaskell.com/chapters)**:** Chapters [4. Syntax in Functions](http://learnyouahaskell.com/syntax-in-functions), [5. Recursion](http://learnyouahaskell.com/recursion), and [6. Higher Order Functions (including Lambdas)](http://learnyouahaskell.com/higher-order-functions).

### [Videos](https://youtu.be/NBKnY7Z_w3I?list=PLPqPwGvHPSZB-urE6QFjKYt6AGXcZqJUh):

[4. Function syntax](http://www.youtube.com/watch?v=vuT8ts_neZw)

[5. Recursion](http://www.youtube.com/watch?v=OfxCm_OarIg)

[6a. Higher order functions](http://www.youtube.com/watch?v=XKUsGSjnITc)

[6b. Lambdas and folds](http://www.youtube.com/watch?v=1IjBT9TSTyQ) (only the lambdas part for this project)

[Point free notation](https://www.youtube.com/watch?v=Cy7jBYr3Zvc) (Not by Peter Drake)

**Part 1.** Write (or find) a Haskell program that uses the [Luhn algorithm](https://en.wikipedia.org/wiki/Luhn_algorithm) to validate credit card numbers. There are many solutions on the web such as [this one](https://docs.google.com/viewer?url=http%3A%2F%2Fozark.hendrix.edu%2F~yorgey%2F490%2Fstatic%2FHaskell-intro.pdf). Your assignment is to create or select one and explain it to me.

**Part 2.** Understand the following solution to the credit card problem and explain it to me. Many of the functions are expressed in “point-free” form (without parameters). Does that mean that those functions don’t expect parameters? If not, what does it mean?

toDigits :: Int -> [Int]

toDigits n = map (\c -> read [c]) (show n)

cycle12 :: [Int]

cycle12 = cycle [1,2]

doubleEveryOther :: [Int] -> [Int]

doubleEveryOther ds =   
 zipWith (\*) ds (if (even (length ds)) then (tail cycle12) else cycle12)

sumDigits :: [Int] -> Int

sumDigits = sum . concat . map toDigits

checkSum :: Int -> Int

checkSum = sumDigits . doubleEveryOther . toDigits

isValid :: Int -> Bool

isValid n = checkSum n `mod` 10 == 0

testCC :: [Bool]

testCC = map isValid [79927398713, 79927398714] -- => [True, False]