CSE 262: Programming Languages

Fall 2016

Homework 2: **Due on September 22nd at 11pm on CourseSite.**

Additional Readings: Learn You a Haskell Chapters 4 and 5. Real World Haskell Chapter 2.

Notes: Don't go crazy about declaring function types, hopefully they'll help you understand how things behave.

1. (5 points): Real World Haskell: Chapter 1, Exercise 2.
2. (7 points): Real World Haskell: Chapter 1, Exercise 3.
3. (8 points): Real World Haskell: Chapter 1, Exercise 4.
4. (5 points): Write a Haskell function that takes one parameter that adds 5 to that parameter. Call this plusFive.
5. (10 points): Write a haskell function that takes one parameter that adds 10 to that parameter using plusFive to accomplish this. Call this plusTen.
6. (10 points): Write a haskell function that takes two parameters and solves for the square root of the sum of the squares of the parameters (distance formula). Call this distance.
7. (10 points): Write a haskell function that takes two parameters, if the first parameter is odd, return first + second. Otherwise, return first - second. Call this decideModify.
8. (10 points): Write a haskell function that takes two list parameters. Return a list that contains the first element of each of the parameters. Call this bothFirsts.
9. (10 points): Write a Haskell function that takes one parameter with recursion that does the following: As long as the input is greater than 0, add 2 and the recursive call of the function with input - 1. This function doubles the input, very poorly but it does. You can assume

it returns 0 when the input is 0 or less. Call this badDouble. The function type is tricky here.

10. (10 points): Write a haskell function that takes one list parameter and reverses the order of the list, without using the built-in reverse function. Call this myReverse.

11. (15 points): Write a haskell function that takes one list parameter. Using the last element of the list (assume non-empty), append last element + 1 to the end of the list. Keep doing this, forever. You will need to use 'take' to view the results of this. Call this foreverIncrement.