

# CS 3035, Fall 2022

## In-Lab Exercise 20

Due on: Nov 2, 2022 (11:59 PM)

This lab involved practicing how to write simple functions in Haskell. For this lab, you do not need to write function headers, i.e., explicit type declarations. For each question, please copy and paste your function expression and the output when its called in the textbox provided for submission. Please state the question number neatly against your answers.

NOTE that this is not the usual way in which we submit labs. Please be mindful of this altered format.

If required, review Lecture18 on Haskell for this lab.

1. Write and call function `takeFromList` that accepts 2 inputs, a list and a number `n`, and it outputs the element at position `n` in the input list. Note that '!' is used to get a list element by its position in Haskell.
2. Write and call a function that takes two parameters, `l` and `w`, and calculates the area of a rectangle with that length and width
3. Write and call a function that takes an int parameter and returns a list of all integers from 1 to the parameter. You do not need to make this work for negative integers.
4. Write a function that takes three numbers, `a`, `b`, and `c`, and returns True if `c` is the square root of (`a` squared plus `b` squared). Haskell has a built-in square root function, `sqr`. For example, `sqr 4` returns 2. Note that `==` is a test, just as in Java, and returns a boolean.
5. Write and call a function that takes three parameters, `a`, `b`, and `c`, and returns True if `a`, `b`, and `c` are a Pythagorean triple (ie if they are possible lengths for the sides of a right triangle), otherwise False. The function must return True if the parameters are a Pythagorean triple, irrespective of the order in which they are given; for example, it must return True for 3 4 5 as well as for 3 5 4, 4 3 5, 4 5 3, 5 3 4, and 5 3 4. Use the `||` logical or operator to string together the cases in which you should return True.
6. Write and call a Haskell function that returns the factorial of its single parameter. Use a range and the product function.
7. Create a list that repeats the names of the Three Stooges (Moe, Larry, Shemp) in order infinitely many times. Write and call a function that takes two parameters, a list and a number, and uses a list comprehension to print "slap" plus each string in the list until `n` slaps have been reported. Here is sample output:

```
slaps 20
["slap Moe","slap Larry","slap Shemp","slap Moe","slap Larry"]
```

8. Write a Haskell function whose content is a list comprehension that returns a list of lists. The return list is a full list of all Pythagorean triples than consists entirely of integers between 1 and 15. It is OK that the GHCI will convert the ints to doubles. The list should not contain duplicates (ie, if it contains [3.0, 4.0, 5.0] it should not also contain [4.0, 3.0,

5.0]) Put all the necessary code in the window, including code you wrote in the ungraded exercise if you reuse it here. Show your output as well.

Hint:

First write code to create a list of all three-member lists [a, b, c] whose members are all integers between 1 and 15

Then refine this code so that the list contains only those lists [a, b, c] for which  $a \leq b$  and  $b \leq c$

Then refine that code so that the list contains only those lists [a, b, c] which are Pythagorean triples.