

## CSCE 314 [Sections 500, 501] Programming Languages – Spring 2020

Hyunyoung Lee

### Homework Assignment 3

Assigned on Tuesday, February 11, 2020

Electronic submission to eCampus due **at 10:00 p.m., Friday, February 21, 2020**

*By electronically submitting this assignment to eCampus by logging in to your account, you are signing electronically on the following Aggie Honor Code:*

**“On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment.”**

In this assignment, you will earn total 100 points. Here are some general instructions.

1. Read the problem descriptions and requirements carefully! There may be significant penalties for not fulfilling the requirements.
2. Some problems ask you to explain the working of your function with the given input. Your explanation must be consistent with your definition of the function. Your work will be graded not only on the correctness of your answer, but also on the consistency and clarity with which you express it.
3. This homework set is an *individual* homework, not a team-based effort. Discussion of the concept is encouraged, but actual write-up of the solutions must be done individually and your final product – the code as well as the comments and explanations – should never be shared.
4. Submit electronically exactly one file named *YourLastName-YourFirstName-hw3.hs*, and nothing else, on eCampus.tamu.edu.
5. Make sure that the Haskell script (the .hs file) you submit compiles without any error when compiled using the Glasgow Haskell Compilation System (ghc or ghci) version 7 and above<sup>1</sup>.

If your program does not compile, you will receive very few points (more likely zero) for this assignment. To avoid receiving zero for the entire assignment, if you cannot complete defining a function correctly without compile error, you can set the function definition `undefined`, see the skeleton code provided.

6. Remember to put the head comment in your file, including your name, UIN, and *acknowledgements of any help received* in doing this assignment. Again, remember the Honor Code!

---

<sup>1</sup>Version 7 is installed in the departmental servers (linux.cse.tamu.edu and compute.cse.tamu.edu), and version 8 is what you will get if you install the Haskell system in your computer.

Below, the exercise problems are from the Haskell Textbook: “Programming in Haskell, 2nd Ed.”, by Graham Hutton. Some problems are modified (with additional requirements) by the instructor. Please read corresponding textbook chapters and the problem statements carefully, paying attention to the requirements. There may be significant penalties for not fulfilling such requirements.

Keep the name and type of each function exactly the same as given in the problem statement and the skeleton code. Also, do not remove or modify the test list or the main function. If you remove or modify those, then you will be penalized for that.

**Problem 1.** (5 points) Put your full name, UIN, and *acknowledgements of any help received* in the head comment in your .hs file for this assignment.

**Problem 2.** (10 points) Chapter 8, Exercise 1. (The function definition for `mult` is given in appendix A.) Carefully study the recursive type of natural numbers in Section 8.4. Using the definitions of the recursive data type `Nat`, `mult` and `add`, show how  $2 \times 2 = 4$  proceeds, in the same way as showing how  $2 + 1 = 3$  proceeds given in Section 8.4 (page 97).

**Problem 3.** (25 points) [Make sure you read Chapter 16 before attempting this problem.] Chapter 16. Exercise 6, page 247. This problem has two parts. Given the following data type

```
data Tree = Leaf Int | Node Tree Tree
```

1. ( $5+5 = 10$ ) Given a tree, function `leaves` counts the number of leaves in the tree, and function `nodes` the number of internal nodes in the tree. Define `leaves` and `nodes`. The function types are as follows.

```
leaves :: Tree -> Int
nodes  :: Tree -> Int
```

2. (Base case 5 points + inductive case 10 points) Prove the following property by induction on trees.

```
leaves t = nodes t + 1
```

**Problem 4.** (60 points) Chapter 8, Exercise 9. Study Section 8.7 carefully before attempting this problem.

Have fun!