

CS330 Spring 2019

Assignment 1

Due on **January 20th, end of the day**. Please follow the submission instructions in the “notes for all labs” on Moodle. You are required to do all of this assignment, but the bolded parts are what you should submit.

- Read the syllabus, linked on Moodle. You are responsible for the contents of the syllabus.
- Read the “Notes for All Labs” document on Moodle. All assignments must be submitted as specified there.
- Read Chapter 1 of your text. This assignment follows along with the material covered in that chapter. Chapter 2.1-2.4 also contains useful information about programming in Haskell that might help if you get stuck.

Written Assignment 1.1: (4 points)

In about one sentence each (in your own words), define the following terms, as described in your textbook:

Function; type; value; programming paradigm

-
- Download and install the Haskell Platform on your machine of choice. This comes with GHCi. It is available on the Lovelace machines, but if you usually work on another machine then you should endeavor to get it installed there. Follow the instructions at <https://www.haskell.org/platform/>
 - Strictly speaking, at this point you only need GHCi, not the entire Haskell Platform, but you might as well get it now, as we will use it over the course of the term.
 - Once you’ve got it running, type in a few basic arithmetic commands, like
 - `4 + 5`
 - `3 / 2`
 - `(2 / 3) + 1`
 - You will notice that `(3 / 2)` in Haskell is 1.5, not 1, as it is in some other languages you might be familiar with. Like Python 3, Haskell will return a float when a division operation would produce a non-integer number.

Written Assignment 1.2: (4 points)

In about one sentence each, describe what all of the parts of this Haskell code are and what they mean and do:

```
size :: Integer
size = 12+13
```

Written Assignment 1.3: (2 points)

In a few sentences, describe what **function composition** is.

Go to <https://wiki.haskell.org/Editors> and select an editor, or see if your favorite editor supports Haskell. I recommend Atom if you are not certain which one to choose. It is a modern, friendly, free, robust editor that works on any platform.

You should also consider visiting <https://wiki.haskell.org/IDEs> to see how you can customize your editor to make programming in Haskell as pleasant as possible. If you don't have experience setting up an editor, it's worth investing the time now, because it will make things smoother later.

Atom instructions: You can get to the package installation window in Atom by hitting `Ctrl+,` and then selecting "install." If you install the packages named under "Atom" on the IDEs page linked above, you will notice that any `.hs` files you create will automatically color the text.

```
1  -- This is a comment. The IDE has made it gray and
2  -- italics on its own, because I installed some packages!
3
4  {-
5   You can see that these are nice and pretty as well,
6   with different types of things each getting their own
7   color.
8 -}
9  square :: Integer -> Integer
10 square n = n * n
11
12 pi :: Float
13 pi = 3.14159
14
15 {-
16  These packages do a variety of other things as well,
17  which we might explore over the course of the term.
18 -}
19
```

Coding Assignment 1.4: (4 points)

Create a new `.hs` file called `your_id_330_lab1.hs`, where `your_id` is your Earlham user ID. All of the functions for the rest of this lab should be in this file, which you should submit along with the rest of your work.

In that file, define at least three functions. All three should be unary functions. (That is, they should take one parameter.) The first two should be regular functions. The third, however, should use a function composition (p. 13) of the first two. They don't have to be fancy. Simple arithmetic functions are fine, but if you can think of something cool to do with what you've read so far, you should go for it. You can also define some values.

Describe what your functions do in a comment or set of comments.

Test your functions and values by loading your .hs file in GHCi or WinGHCi, and calling them to see how they behave. If you get stuck, you can read into Chapter 2 or ask for help on Piazza.

Written Assignment 1.5: (3 points)

In a few sentences, describe **types** and **type checking**. In a few sentences, describe what **type abstraction** means.

Written Assignment 1.6: (4 points)

In about a paragraph, describe at least two ways in which Haskell differs (or seems to differ) from another programming language you like, and at least two ways in which it is alike. If you're having trouble coming up with things, make sure you've read through chapter 1. (This will only be graded on completeness; I don't expect you to delve into Haskell's theoretical underpinnings at this point to verify your answer.)

Written Assignment 1.7: (2 points)

In a few sentences, describe your current understanding of what functional programming is. (This will only be graded on completeness; try to describe it from your own understanding, rather than by just googling a definition.)

Coding Assignment 1.8: (4 points)

In the same .hs file as from before, write a few functions that do some basic things, in addition to the ones you wrote earlier. Load your .hs file in GHCi, and try calling the functions with different values in the interactive window.

Discussion Prep 1.9: (4 points)

List at least three thoughts or questions you had about the chapter 1 reading.

Preparation for the Future

Moving forward, the expectation will be that you feel comfortable with the topics covered in this assignment. If you do not, you should ask questions on Piazza and/or continue to practice with GHC.