

Lab Sheet 1

Starting with Lists

Functional Programming - Winter 2018/2019 - October 25, 2018 - Schupp/Lehmann

How to succeed with the labs and exercises?

Labs and exercise sheets are published every week on the course homepage at StudIP. As described in the first lecture, each successfully completed lab and exercise earns you bonus points towards your final score in this semester's exam. Keep in mind that you only get bonus points if you would pass the exam without them. **Cheating does not help you - but we will!**

How to complete a lab successfully?

In the lab, you will solve the lab sheet with fellow student(s). To find out who your teammates are, please look at the group pdf in StudIP's download area. **During the session, one participant of every group will be selected, who must then explain one of their task solutions to a lab assistant.**

How to complete an exercise successfully?

In order to complete an exercise sheet successfully, you must upload your answers using CAT on StudIP **before the deadline** printed on the exercise sheet. We will not consider any solutions handed in after the deadline! Furthermore, you must solve and hand in the exercises **individually** and your Haskell code **must compile** and **pass certain amounts of tests** as specified. During the exercise session, we develop possible solutions together. Please participate! We encourage you to ask and answer questions from fellow students.

Technically, Haskell files you submit using CAT on StudIP must have the format as specified by CAT (usually ".hs", ".lhs", or ".txt"). Furthermore, CAT will store your last submission only. Therefore, if you first submit successfully (your code compiles and tests are passed) and afterwards unsuccessfully (your code does not compile or certain tests fail again), your last submission counts, and - if it does not compile - will therefore be ignored. Make sure your last submission was successful!

How to get additional information?

We encourage you to discuss past and present exercise sheets with us. Either approach us during the exercise session, or visit us during the weekly office hours. We are also available via e-mail or on the StudIP forum. We try to reply as quick as possible and in general, you should get a reply the next weekday, but we cannot guarantee this.

Getting Started with Haskell Platform/WinGHCi

Add “Haskell” using “MyApps”. Launch “WinGHCi” from Start Menu.

Optional: You may also want to download the Atom text editor. Visit <http://www.atom.io>, click on “Other platforms”, then download and unpack “atom-windows.zip”. Afterwards, start Atom and install the “language-haskell” package in order to get Haskell syntax highlighting.

Playing with Lists

Lists in Haskell are denoted by brackets and the values in the lists are separated by commas.

Exercise 1 To solve this exercise you should only use the functions `==`, `head`, `init`, `last`, `tail`, `drop`, `reverse`, and `take`. Consider the following list of characters

```
['n', 'u', 'f', 'F', 'P']
```

and answer the following questions:

Hint: First find out what the functions do¹!

- a) What is the return value of the expression `take 3 ['n', 'u', 'f', 'F', 'P']`?
- b) What is the return value of the expression `reverse ['n', 'u', 'f', 'F', 'P']`?
- c) Is the list `['n', 'u', 'f', 'F', 'P']` the same as the string `"nufFP"`?
Hint: Use `==` to test for equality!
- d) How can you get `'n'` as return value?
- e) How can you get `'P'` as return value?
- f) How can you get `"ufFP"` as return value?
- g) How can you get `"FP"` as return value?
- h) How can you get `"fun"` as return value?

Exercise 2 Given two lists, `[4, 1]` and `[3, 2]`. Provide a Haskell expression that returns the list `[1, 2, 3, 4]`. Try the following functions and operators: `head`, `tail`, `!!`, `++`, `take`, `drop`, `reverse`, `init`

Exercise 3 Provide a Haskell expression that returns a list containing all the integers between 1 and 99 that are divisible by 3. **Hint:** There exists a short form for creating lists of consecutive numbers.

¹<http://www.haskell.org/hoogle> is a good² place to look that up!

Exercise 4 *

- a) Do both the symbols `[]` and `[[]]` represent the empty list? Motivate your answer.
- b) Enter the following expressions in GHCi and observe the result. Why does GHCi return what it does?
 - a) `reverse []`
 - b) `head []`
 - c) `sum []`