

Lab Task 1: Getting started with Haskell

Checklist Physical Lab: Swiped your student card? Your lab space has a tick? Cleaned your space, keyboard, mouse with a wipe? Facemask on?

Checklist Online Lab: Please register with your official name. Have you already installed Haskell on your computer?

General instruction – University labs: Windows machine: Open the ZENworks application. If the Haskell icon does not already show, search for Haskell in the box on the top, and click on it. This should install Haskell locally on the computer. [This could take a while when done the first time, just start reading through Task 1.1 meanwhile.] Clicking again should then start the Glasgow Haskell Compiler `ghci` [the “i” stands for interactive] . You should see the prelude prompt.

Try first with evaluating simple expressions in Haskell (Task 1.1.):

Task 1.1. Using Haskell as calculator After starting `ghci`, try some of the examples given in the course. What is the value of:

- a) arithmetic operations: `5+4`, `4/3`, `4==3`, `15 ‘mod‘ 6`
- b) operations on lists: `head [1,3,5]`, `drop 4 [1..10]` .
- c) What do you get if you call `tail` with the list of digits in your student number?
- d) You can store the result temporarily with a `let` construct: `let result = 3 + 5` and then call `result`. Challenge question: can you also define functions with a `let` construct?

The demonstrator later will ask you one of the above questions, and you should be able to answer this question.

Next, to be able to write programs, you need to create a file stored at a suitable place. I recommend to create a folder for CS-205 and a subfolder for Lab1.

Task 1.2. Load first program Create a file called `double.hs` using your favorite editor. [Doublecheck that the file is not accidentally called `double.hs.txt`] and type the double function.

```
doubleMe :: Int -> Int
doubleMe x = x + x
```

To load an existing Haskell file, there are three options, the first one is probably the easiest:

1. click on the file icon, and it will open `ghci` with the file loaded.

2. at the prelude prompt, type `:l <full path> double.hs`
3. go to the directory with the file directly in haskell using the `:cd` command.

Once the program `double` is loaded, check whether it computes the desired result (at the `ghci` prompt call e.g.:

`doubleMe 6`).

- b) What is the result of `doubleMe (doubleMe <your studentnumber>)`? Include the Haskell call and response as a comment (comments start with `--`) in your file.
- c) Add a function `doubleUs :: Int -> Int -> Int` that takes two arguments, say `x` and `y`, doubles both, and adds them. Check again whether it computes the correct result when called at the `ghci` prompt.
- d) Write a second version of this function, called `doubleUs2`, which does the same, but makes use of the function `doubleMe`.
- e) Also define a function `quadrupleMe` which applies `doubleMe` twice. [This function applied to your student number should give the same result as above.]

Task 1.3. Use list operations from the second chapter, (i.e. `head`, `tail`, `!!`, `take`, `drop`, `length`, `sum`, `product`, `++`, `reverse`) to define the following further operations/functions on lists. (`[Int]` is the type of lists of integers).

- a) `second :: [Int] -> Int`, computing the second element of a list.
- b) `last2 :: [Int] -> Int`, computing the last element of a list.
- c) `init2 :: [Int] -> [Int]`, computing the initial segment of a list, respectively.
- d) `last3 :: [Int] -> Int`, computing the last element of a list, in a different way, i.e. using different operations.
- e) `init3 :: [Int] -> [Int]`, computing the initial segment of a list, in a different way.

You may assume in this first lab session that the functions are only called with lists which have at least two elements. I.e. the result, when run in Haskell with a list `[3,4,6,7]` should be: (But please try other inputs as well.)

```
> second [3,4,6,7]
4
```

```
> init2 [3,4,6,7]
[3,4,6]
```

Before you leave make sure that you have spoken to a deminstrator, and the demonstrator has signed you off. There will be a catch up opportunity next week for those who started late or were unwell today. But it is in your interest to start this week and get as far as you can. Please also do not share any solutions, but give every student the opportunity to solve the tasks themselves.

Have fun!