

- Mark your completed exercises in the OLAT course of the PS.
- You can use a template .hs file that is provided on the proseminar page.
- Upload your modified .hs file for Exercise 2 in OLAT.
- Your .hs file should be compilable with ghci.

**Exercise 1** *Haskell setup***5 p.**

Set up a working Haskell environment on your computer and get familiar with `ghci`. To do this follow these steps:

- Install Haskell, e.g., via `ghcup`.<sup>1</sup>
  - Run `ghci` in a terminal and evaluate the expression  $(5 + 2) * 3$ .
  - Find and install a suitable text editor for your system to write and edit .hs files.<sup>2</sup> You can try one of the following free editors:
    - Notepad++<sup>3</sup> (Windows)
    - Gedit<sup>4</sup> (Windows, macOS, Linux)
    - Visual Studio Code<sup>5</sup> (Windows, macOS, Linux)
  - Copy or enter the following code in your text editor and save it to a file called `myProgram.hs`. Be sure to use standard double quotes (`"`), but neither two single-quotes (`'`) nor fancy-looking double-quotes (`“` or `”`).
- ```
hello :: String -> String
hello xs = "Hello " ++ xs
```
- Load the file in `ghci` with the command `ghci myProgram.hs`
  - Evaluate the expression `hello "World"`
  - Make yourself familiar with `ghci`. In particular, try the following commands:
    - `:?` – help
    - `:load name.hs` or `:l name.hs` – load Haskell script `name.hs`
    - `:reload` or `:r` – reload current Haskell script
    - `:edit` or `:e` – edit current Haskell script
    - `:set editor someEditor` – set `someEditor` as preferred editor

Further investigate what happens if you type `h` and then the tabulator key, or `hel` and the tabulator key.

You can find links to introductory material about installing Haskell, the command line, etc. on the lecture homepage.<sup>6</sup>

<sup>1</sup><https://www.haskell.org/ghcup/>

<sup>2</sup>Word processors like Microsoft Word, Apple Pages, ... are not text editors.

<sup>3</sup><https://notepad-plus-plus.org/>

<sup>4</sup><https://wiki.gnome.org/Apps/Gedit>

<sup>5</sup><https://code.visualstudio.com>

<sup>6</sup>[http://cl-informatik.uibk.ac.at/teaching/ws23/fp/ghc\\_setup.php](http://cl-informatik.uibk.ac.at/teaching/ws23/fp/ghc_setup.php)

## Exercise 2 *Writing simple functions*

5 p.

1. Define a function `volume r = ...` to compute the volume of a sphere with radius `r`. (1 point)
2. 400 liters of helium cost 50.99 EUR. Define a function `heliumCosts` that computes the costs (in EUR) of filling a balloon with radius `r` (in cm). Note that 1 liter is 1 cubic decimeter.  
Hint: It might be worth to write further auxiliary functions to split up the task. (1 point)
3. Define `balloonRadius` as the inverse function of `heliumCosts`: it takes some amount of money `m` in EUR and computes the radius `r` of the balloon (in cm) that you can fill with `m` EUR.  
Hint: `x ** (1/3)` computes  $(x)^{\frac{1}{3}} = \sqrt[3]{x}$ . (2 points)
4. Test that `balloonRadius (heliumCosts r)` is `r` and `heliumCosts (balloonRadius m)` is `m` for some test values of `m` and `r`.  
If you did not solve `balloonRadius`, just make sure that your implementation of `heliumCosts` is plausible, e.g., `heliumCosts 20` is 4.27 EUR, and `heliumCosts 10` is 0.53 EUR. (1 point)