

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

Exercise 1 *Parsing expressions***5 p.**

Construct the abstract syntax trees for the given expressions:

1. `2 * (3 + 1)` (1 point)
2. `(x > 3) && (y == (7 - 2)) || (z >= 4)` (2 points)
3. `cube (4 + 1) * (height * width * depth)` (2 points)

Remark: Function applications (e.g., `cube 4`) bind stronger than operator applications (e.g., `8 * 3`). Also note the precedence rules for logical operators: `&&` has higher precedence than `||`.

Exercise 2 *Datatype definitions***5 p.**

In this exercise you should design datatypes for listing objects in a fridge. You can use the Haskell template provided on the course website for this exercise.

1. Each object in a fridge has a name and an expiration date. Moreover, each object either has a quantity, e.g., a box of 6 apples, or it is a fluid that has a volume, e.g., 0.5 liters of juice.
Define a datatype in Haskell called `FridgeObject` to represent such a fridge object. Of course, you may also define auxiliary other datatypes. (1 point)
2. Define the following fridge objects in your Haskell program: (1 point)
 - (a) a box of 4 apples with expiration date October 31, 2023
 - (b) 2.3 liters of milk with expiration date August 4, 2023
 - (c) a net of 6 lemons with expiration date November 3, 2023
3. Define a datatype `FridgeList` that represents a list of `FridgeObjects`. (1 point)
4. Consider an example fridge that contains two boxes of apples as specified in (a), and the amount of milk as specified in (b). (2 points)
 - Draw the tree that corresponds to the list of objects of the example fridge.
 - Define a constant `exampleFridgeList` in Haskell that represents this tree.
 - Is the representation unique?