Name: _____

Matriculation Number: _____

Points: _____

This exam consists of 2 exercises, for a total of 22 points (so there is roughly 1 point per 2 minutes).
The available points per exercise are written in the margin.

Exercise 1: Multiple Choice 12

In each multiple choice question, you get

- 3 points for 3 correct answers,
- 1 point for 2 correct answers,
- 0 points, otherwise.

Consider the following Haskell function.

```
foo x y z
  | null x = y
  | null z = y
  | otherwise = foo (tail x) ([head x] ++ y ++ [head z]) (tail z)
```

(a) Mark all valid evaluations. (3)

☐ `foo [1] [2] [3] = [1,2,3]`
☐ `foo [1,3,4] [2] [3,4] = [4,3,2,3,4]`
☐ `foo [1,0] [2] [3,4] = [0,1,2,3,4]`

(b) Mark all valid type declarations of `foo`. (3)

☐ `foo :: (Num a) => [a] -> [a] -> [a] -> [b]`
☐ `foo :: String -> String -> String -> String`
☐ `foo :: (Fractional d) => [d] -> [d] -> [d] -> [d]`

(c) Mark all valid type declarations of `foo`. (3)

☐ `foo :: [Integer, Integer, Integer] -> [Integer]`
☐ `foo :: [Integer] -> ([Integer] -> [Integer]) -> [Integer]`
☐ `foo :: [Integer] -> ([Integer] -> ([Integer] -> [Integer]))`

(d) Mark all declarations that compile without errors. (3)

☐ `bar x = [1+2+3] ++ x`
☐ `com x y = if x == y then 1 + (com x) ((+) 1 y) else 0`
☐ `data Foo b a = Bar String | Foo Char [a] [c]`

Exercise 2: Programming 10

(a) Define a datatype `Volume` with three constructors corresponding to a tablespoon, a teaspoon and a cup. (1)

(b) Define a function `metric` which converts volumes into their value in liters. Give both the type definition and the defining equations. Assume the FDA definitions (one teaspoon is $5ml$, one tablespoon is 3 teaspoons, one cup is $0.24l$). (3)

(c) Write a type-class instantiation such that `Volume` becomes an instance of `Ord`, ordered by their represented volumes. You can assume that the function `metric` from the previous exercise exists. (3)

(d) Define a function `contains` which takes a function `Volume -> Bool` and a list of `Volume`s, and which returns `True` if there is an element in the list for which the supplied function returns `True`. Give both the type definition and the defining equations. (3)

- You can use the predefined functions `head`, `tail`, and `null`.
- It is not allowed to use any other predefined function on lists.
- It is not allowed to use pattern matching.