CCPS506 Lab 5 – Haskell: Functions, types, type signatures

Preamble

Haskell is a statically typed language, yet we are not required to explicitly assign type when naming variables and functions. Haskell instead uses an inference engine to decide, at compile time, what type (or type class) a value or function will have. We can ask Haskell to report this type in **GHCi** using **`:t`**.

It is common in Haskell to describe functions using this type signature, so it's important to understand them. In this lab, type signatures are provided for several functions that you will write. When your function is loaded and we ask **GHCi** for its type using **`:t`**, it should match the description.

Getting Started

Create a Haskell source file called **`Lab5.hs`**. In this file you will define a module named **`Lab5`**. All functions written for this lab will be placed in this single module. Make sure each of your functions is named precisely according to the specifications below.

Lab Description

Write a Haskell function to solve each of the problems below. Your function's type signature must match what is indicated. Don't fret – if you solve the problem in the first place, there's a very good chance that your signature will match. You should still verify this, however.

i) **`thirdLast :: [a] -> a`**

Returns the third last item in an input list. You may assume the list has at least three elements. You may NOT use the **!!** operator.

ii) **`everyOther :: [a] -> [a]`**

Returns a list containing every other element, starting with the first element. Your function should work regardless of the size of the list. It could have an even or odd length. You may assume the input list is not empty. You may NOT use the **!!** operator.

iii) **`sumPosList :: (Num p, Ord p) => [p] -> p`**

Returns the sum of all positive values in a list. The sum of the elements in the empty list is zero.

Submission

Labs are to be completed and submitted *individually*. Submit your **Lab5.hs** file containing the **`Lab5`** module and all completed functions on D2L, under the submission for Lab 5.