School of Computer Science, University of Nottingham

G51PGP Programming Paradigms, Spring 2019

Graham Hutton

Haskell Coursework I, Part 2/3

Deadline: Wednesday 20th February 2019, 12 noon

This exercise sheet covers the material from Lecture 3 (Types and Classes), and is worth 2% of the overall module mark. Half of the marks are awarded for your solution, and the other half are awarded for your answers to the tutor's questions. You should attempt to complete the exercises in the sheet in your own time, but if you get stuck you can ask for help from the tutors during the weekly lab sessions.

Assessment will be carried out by oral examination during the lab session – nothing needs to be handed in. When you have completed the exercises ask a tutor to examine your solution. The tutor will then ask you some questions to test your understanding. Note that tutors will only be available to answer questions and assess your solution during the official G51PGP lab sessions (Wednesdays, 11am to 1pm, A32).

You can check your solution by loading your script into GHCi. If you have made a type error it will be detected and reported by GHCi. I recommend reloading your script after adding each definition, so that any errors can be detected immediately.

Ensure that your script type checks before submitting it for assessment. If you are unable to complete some definitions then comment out those that are incomplete or incorrect. In Haskell, multi-line comments start with `{-` and end with `-}`, while single-line comments begin with `--` and extend to the end of the current line.

1. Fill in the gaps (the question marks `?`) in the following definitions to make them type correct. It does not matter what the definitions do as long as they are type correct. Note: you should type in and complete each definition one at a time.

```
e1 :: ?
e1 = [True,False,True]

e2 :: ?
e2 = [[1,2],[3,4]]

e3 :: (Char,Bool)
e3 = ?

e4 :: [(String,Int)]
e4 = ?

e5 :: ? -> ?
e5 n = n*2

e6 :: Int -> Int -> Int
e6 ? = ?

e7 :: ? -> ?
e7 (x,y) = (y,x)

e8 :: a -> (a,a)
e8 ? = ?
```

2. When you submit your solution to a tutor, you will be shown some definitions and asked to state their types. Before submitting, you should practise by adding types to the following definitions; try to give the most general types possible.

```
nums = [1,2,3,4,5]

table = [(False,1),(True,2),(False,3)]

one x = [x]

three x = (x,x,x)

first x y = x

mult m n = m*n
```