McMaster University
Department of Computing and Software
Dr. Wolfram Kahl

SE 3E03, Fall 2002
Assignment 5

Design and Selection of Programming Languages

26th September 2002

Problem 19

Work through lessons 1–11 from the "Two Dozen Short Lessons in Haskell" by Rex Page. Work through lessons 1–4 and 7 **before the lecture on Monday!**

With the command

```
source /u2/se3e03/admin/etc/.cshrc
```

issued interactively or included in your `$HOME/.cshrc`, you should have the Haskell interpreter `hugs` avaliable on all departmental SUNs.

Problem 20 (HTML in Haskell)

In Haskell one can define **type synonyms** (similarly to Oberon); the following example defines a type for a pair consisting of an integer and a string:

```
type MyPair = (Integer, String)
```

a) Define a type synonym `STag` for the information contained in HTML start tags, as analysed in Problem 17.

b) Find out which function the prelude function `unwords` implements. (**Hint:** Find out its type first!)

c) Implement a function `showSTag :: STag -> String` that converts values of type `STag` into correct HTML start tags.

Problem 21 (Parsing in Java — until October 10)

Extend your solution for Problem 13 with a recursive descent parser as shown in the lecture.

a) Collect all the requirements for the lexing component.

b) What options do you have for how to design the class structure to include lexing and parsing functionality?

c) Implement the lexing and parsing functionality for your arithmetic expressions.

d) Implement a command-line application for evaluating expressions and modifying environments. Document questions and design decisions that arise.