Effective Functional Programming
Prelude
Assignment 0
`Hello, World!`

Paul Downen

1 All About You (10 points)

Exercise 1.1 (10 points)**.** Write a short description on your previous experience with functional programming, and what you hope to get out of this class. Have you written a program in a functional language (like Haskell, OCaml, SML, Scheme, Clojure) before? If so, which one and what program? Or have you used a functional programming technique (such as MapReduce) in some other setting? How comfortable are you with functional programming basics (like recursion, higher-order functions, generics (a.k.a. parametric polymorphism), immutability, etc.)? At the end of the term, what do you hope to learn that you didn't know before? **End Exercise 1.1**

2 Let's Get Hacking (10 points)

Exercise 2.1 (10 points)**.** Download and install Haskell on your system (links can be found in the Resources page of the class site). I suggest either installing the GHCup `https://www.haskell.org/ghcup/` or the Haskell Platform `https://www.haskell.org/platform/`. To use GHCup to set up your system, use a terminal/command prompt to run the command

```
> ghcup tui
```

for a text-based interface where you can select which tools to download and install (you should download at least the recommended versions of `GHC` and `cabal`; optionally you can install `Stack` and `HLS` as well).

Cabal is a tool that helps you automatically download and use packages off of the online Hackage database. Once you have installed Cabal from the above step, to get it up running, use a terminal/command prompt to run

```
> cabal update
```

which downloads a copy of the current list of Hackage packages to your computer.

Haskell Stack is a tool that helps automatically manage projects for you. Get the Haskell Stack set up by running the command

```
> stack setup
```

Running Haskell: Once you have completed the above steps sucessfully, you can compile and/or run Haskell programs on your computer! Open up the Haskell interpreter `GHCi` in a terminal/command prompt with the command

```
> ghci
```

Or alternatively, if you have `Stack` installed, you can run

```
> stack repl
```

Now at the interpreter prompt, type in the following expression exactly as shown and press enter:

```
let motor sounds = sounds "Vroom! " in motor cycle
```

Describe in words what happens. **End Exercise 2.1**

Hint 2.1. Holding the "control" key and pressing "c" will halt GHCi. *End Hint 2.1*