

Notes from 11/9

Announcements

Final exam *was* scheduled for December 21st. That's very late. Instead, we'll either:

1. Do it for ~1 hour in class on December 14, or
2. Do a take-home exam where you receive it on 14th, due a few days later.

Will confirm next week.

Modules

- Imports, qualified, renames.
- Selective exports
- Abstract data types by not exporting constructor.
- newtype
- Example: case-insensitive strings

A technique for abstract data types, sometimes called a “smart constructor”. Define a data type, and make the type itself public, but the constructor(s) are private.

Maps and Sets

- qualified import
- where to find docs
- fold

Lists are convenient, but not very efficient.

```
import qualified Data.Map as M
```

```
tally :: Char -> M.Map Char Int -> M.Map Char Int
tally c m =
  case M.lookup c m of
    Nothing -> M.insert c 1 m
    Just k -> M.insert c (k+1) m
```

Input/Output

- IO monad
- putStr[Ln], getLine, print
- System.IO
- withFile, hPutStr, hGetLine, ...

```
main = return ()
```

```
mymain = do  
  putStrLn "Hello, world."  
  putStr "What is your name? "  
  name <- getLine  
  putStrLn $ "Welcome, " ++ name
```