HW 6 **Prof. Caldwell**
Due: 16 September 2008 **COSC 3015**

1

Problem 1.1. Start reading chapter 3 of Bird.
Here is some code having to do with dates.

```
-- Following British convention ... the numbers in a date represent (day,month,year)
data Date = DMY (Int, Int, Int)

leap y = (y `mod` 4 == 0) && not(y `mod` 100 == 0 )
    ||  (y `mod` 4 == 0) && (y `mod` 100 == 0 )   && (y `mod` 400 == 0)

days_in_month y m
   | has_31                     = [1..31]
   | not (has_31) && m /= 2     = [1..30]
   | m == 2 && not (leap y)      = [1..28]
   | m == 2 && leap y            = [1..29]
 where
     has_31 = m `elem` [1,3,5,7,8,10,12]

goodDate (DMY(d,m,y)) = d `elem` (days_in_month y m)

dates_in_year y =     do m <- [1..12]
     d <- days_in_month y m
     return (DMY (d,m,y))

data Months = January | Feburary | March | April | May | June |July
             | August | September | October | November | December
  deriving (Enum, Show)
```

Problem 1.2. This is essentially the harder version of problem 2.7. of Bird pp. 55. Make the type `Data` an instance of the type class `Show` and define a show function that prints dates in the following (ordinal) form. If a date is invalid – you code should raise an exception by calling `error "bad date!"`. Test your program by evaluating the expression `(dates_in_year 1956)` and submitting the output with your code. Note that my show function appends `"\n"` to the end of the date string so that they print on their own lines.

```
Main> DMY (2,1,2008)
2nd January 2008
Main> DMY (29,2,2008)
29th Feburary 2008
Main> DMY (29,2,2007)
Program error: bad date!
```