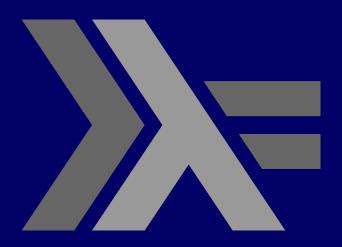
## PROGRAMMING IN HASKELL



Chapter 4.4 – Closures and Partial Functions

## First Class Functions.. recall

```
inc :: Num a => a -> a inc n = n + 1
```

```
double :: Num a => a -> a
double n = n * 2
```

```
square :: Num a \Rightarrow a \rightarrow a square n = n \land 2
```

```
ifEven :: Integral a => (a->a) -> a -> a
ifEven f n =
  if even n
  then f n
  else n
```

However, we are still repeating code

```
*Main> ifEven square 4
16

*Main> ifEven inc 4
5

*Main> ifEven inc 5

*Main> ifEven double 4
8

*Main> ifEven double 5
5

*Main> ifEven square 4
16

*Main> ifEven square 5
5
```