

# List Processing

## Composite Data Types in Haskell

Prof Hans Georg Schaathun

Høgskolen i Ålesund

22nd January 2016

# Outline

Generic definitions — Polymorphism

Finding more functions

# Polymorphism

- `length :: [Integer] -> Integer`
- `length :: [Double] -> Integer`
- `length :: [Bool] -> Integer`
- **Do we have to make our own for `[Customer]`?**

# Polymorphism

- `length :: [Integer] -> Integer`
- `length :: [Double] -> Integer`
- `length :: [Bool] -> Integer`
- Do we have to make our own for `[Customer]`?

No, we have polymorphism

- `length :: [a] -> a`
- `a` is an arbitrary type

## Another example

1. `zip :: [a] -> [b] -> [(a,b)]`

2. Two arbitrary types `a` and `b`

3. `zip ['a'..'f'] [0..]`

→

`[('a',0), ('b',1), ('c',2), ('d',3), ('e',4), ('f',5)]`

# Recursion

## Example

```
length [] = 0
```

```
length (x:xs) = 1 + length xs
```

## Example

```
pair [] = []
```

```
pair (x1:x2:xs) = (x1,x2):pair xs
```

# Overloading

- Polymorphic functions

1. one definition for multiple types
2. e.g.  $\text{fst } (x, y) = x$

- Overloading

1. one function name for different definitions
2. different definitions for different types
3. e.g.  $x == y$

- We will get back to overloading later

# Outline

Generic definitions — Polymorphism

Finding more functions



## Some list functions

<code>!!</code>	<code>[a] -&gt; Int -&gt; a</code>	get element by index
<code>head last</code>	<code>[a] -&gt; a</code>	get first/last element
<code>tail init</code>	<code>[a] -&gt; Int -&gt; [a]</code>	get all elements but the first/last
<code>reverse</code>	<code>[a] -&gt; [a]</code>	reverse order
<code>replicate</code>	<code>Int -&gt; a -&gt; [a]</code>	a list repeating the same element

# Standard libraries

1. Prelude is loaded by default
2. Load other standard modules
  - `import Data.List`
  - `import Data.Array`
3. Install new packages
  - `cabal install easyplot`

# The Haskell Platform

1. Compiler and interpreter: GHC/GHCi
2. Standard libraries
3. Package management tool: cabal
4. C/Haskell interface tool: Hsc2hs
5. Other tools