

# Chapter 3.6, 10, 13

## Reading Discussion

# Functional languages

- Work in small groups
- Don't look at the text or your notes
- List as many functional languages as you can recall
- For each language list some features that that language is known for.

# Features

- What features make a language functional
- What features are not necessarily functional but are found more often in functional languages

# Features

- What features make a language functional
  - First class functions
  - Higher order functions
- What features are not necessarily functional but are found more often in functional languages
  - Types and polymorphism
  - Algebraic datatypes (lists, trees)
  - Garbage collection
  - Pure features (no side effects)
  - recursion

# E3 and Scheme

- The language E3 is based upon Scheme
- List as many features as you can recall from Scheme found in E3
- What features found in Scheme are not in E3?

# E3 and Scheme

- The language E3 is based upon Scheme
- List as many features as you can recall from Scheme found in E3
  - Dynamic typing and predicates (like: char? = ischar, number? = isint, pair? = ispair)
  - Lists as pairs
  - prefix notation and parentheses
- What features found in Scheme are not in E3?
  - First class functions
  - Programs as lists (self representing)

# Normal order evaluation

- What is normal order evaluation?
- What are other choices?
- How do they compare with normal order evaluation?

# Currying

- What is currying?