

# Chapter 6 Reading Discussion

Chapter 6 (pp 217-287)

# Control Flow

- Without consulting the text or your notes
  - Work in groups of 3
  - List as many control flow mechanisms as you can remember
  - What language paradigm
    - Declarative
    - Imperative
    - Logical

Are they sometimes associated with?

- Time 1 minute

# Order of evaluation

- Without consulting the text or your notes
  - Work in groups of 3
    - List
      - Benefits
      - Problems

If order of evaluation of sub expressions is not specified.

# Assignments

- Without consulting the text or your notes
  - Work in groups of 3
  - Issues
    - L-value v.s. r-value
    - References and pointers
    - Side effect causing functions
    - Boxing
    - Expressions v.s. commands
    - initialization
  - Give an example that illustrates why the issue is important

# Precedence v.s. order of evaluation

- Consider  $x * y + z$
- When would  $z$  be evaluated first?
  1.  $(+)$  associates to the left
  2.  $(+)$  associates to the right
  3.  $(*)$  binds tighter than  $(+)$
  4.  $(+)$  binds tighter than  $*$
  5. The language evaluates sub expressions from left to right
  6. The language evaluates sub expressions from right to left
  7. The language specifies no order of evaluation
  8.  $(+)$  is commutative
- Make a list of all conditions that must hold
- Could  $y$  ever be evaluated first?

# Mathematical Laws

- True or False
- $x - y + z$
- Can be rearranged to be
- $(x + z) - y$

# Short circuit evaluation

- Works in groups of 3
- Give an example where short circuit evaluation of (&&) logical conjunction makes code shorter.
- Can you think of one for logical disjunction?

# Multiway case

- Work in groups of 3
- List as many considerations as you can think of that might be important to efficient implementation of multiway branches



# Loops

- Kinds of looping constructs
  1. Enumerated loops
  2. Logical loops
  3. Use of iterators
  4. Mid test loops (break, exit, continue, etc.)
  5. Recursion
- Give a pair  $(m,n)$  where construct  $m$  can simulate construct  $n$ . Justify your answer with a sketch of an implementation.