

(See slides for lecture #2)

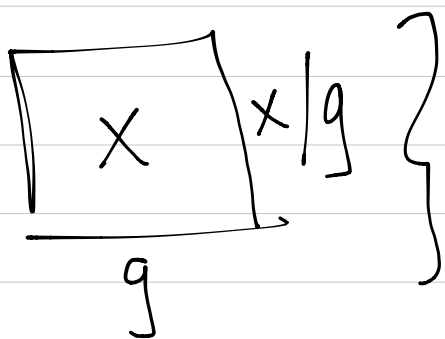
*Applying compound procedures.

**Can <define> the \oplus symbol can be "defined" to be 4

*Church-Rosser theorem. The consistency of mathematical evaluation. "Any order is OK, it produces the same result." Applicative order is used in Scheme (not normal order)

Cursing + Recursing
* Declarative vs Imperative definitions — (what is vs. how-to)
↳ "Good-enough" answer for \sqrt{x} code.

Predicate + consequent.
* $\langle \text{if} \rangle$ has to be a special form

 } Newton's Method

* Gains accuracy w/ each iteration
... Getting closer to \sqrt{n} by $1/2$
each time/iteration. Getting
another bit in binary.

* Naming + the environment

(y and w come from different environments)

↳ Names of formal parameters don't matter, but the names of external variables do matter.

[Expression - evaluates to something
Command - Does something

→ Doesn't change the world.
Completely static. Platonic world.

↳ Freedom of expressions

→ DOES change things! Dead goldfish example