

systems = 6.033 + 6.826

“The MIT Approach”

- **Simplicity:** The design must be simple, both in implementation and interface. It is more important for the interface to be simple than the implementation.
- **Correctness:** The design must be correct in all observable aspects. Incorrectness is simply not allowed.
- **Consistency:** The design must not be inconsistent. A design is allowed to be slightly less simple and less complete to avoid inconsistency. Consistency is as important as correctness.
- **Completeness:** The design must cover as many important situations as is practical. All reasonably expected cases must be covered. Simplicity is not allowed to overly reduce completeness.

Abstraction

what

how

correctness

efficiency

interface

implementation

mathematical

messy

eg, storage

L1, L2, core, disk

some uses of graphs in
computer science

using graphs in cs

- **reachability:**
 - data flow: push values through the graph
 - safety: a bad thing never happens
 - liveness: a good thing eventually happens
- **computation:**
 - neural nets
 - bayes nets
- **visualization:**
 - automatic layout
 - zooming interfaces, etc