Meeting 21: Imperative Computation

This Week
- Type Safety
- Variables: Mutable State
- Pointers: Aliasing

Midterms Returned
Fri in Recitation

Course Design
- Real-world: Tackle open-ended problems
- Whether there is a solution is unclear

Classroom/Homeworks: Rate, uncreative execution tactics

This course - “middle ground”
- Homeworks to drive discussion
- Class: There may be things that aren’t 100% clear yet
Not be completely open-ended by having scaffolding

Extend HW5 to end of week
Set 4/7 + 1 late day period

- define
- infer type
- union + type + records
- name vs. struct equality

\[[I: \text{Int}, B: \text{Boolean}] = \text{Foo}\]

abstract class Foo
case class I(i: Int) extends Foo
case class B(b: Int) extends Foo
\( m : [E : \text{Int}, B : \text{Boolean}] (I : 2) \approx I (2) \)

\[
\text{if (true) } e_2 \text{ else } e_3
\]

Posted Screencast

- Walkthrough doing a 
  
  part HWS - GetField

\[
g() \text{ match } \$ \\
\text{ case } I : x \Rightarrow \\
\text{ case } B : y \Rightarrow \\
3
\]