Announcements

Office Hours
- Evan Tue 10:45-11:45, Thu 1-2 ECCOT 621
- Huck Mon 10:30-11:30 ECCS 122 (CSEC)
- Aleks Mon 3-4 ECCR 1BO9
- Yi-Fan Wed 10:30-11:30 ECCR 1BO9
- Sam Wed 3-4 ECCR 1BO6

Recitation 9-9:50 Room Change ECCR 1BO9
Bring laptops to section — getting started w/ Scala

Next Week
- Evan traveling
- Tue Huck/Aleks
- Thu - No class — Video
  - HW1 clue 1/30
Notes + Class supplement each other

Anyone read editorial?

Imperative vs. Functional "Schism"

- mutable state
  - "executing statements" modifying memory

```plaintext
int x;
int y;

x = 3;
y = 4;

-> code
```

```plaintext
X = y
```

```plaintext
3 0x1
10x1
Ox1 4
```

```
mem
```
"execute for effects"

(side-effect on memory)

Functional World

"evaluating expressions"

\[(1+1) + (1+1) \rightarrow 2 + (1+1) \rightarrow 2 + 2 \rightarrow 4\]

"pure" = "effect-free"
interacting w/ external
- printing

is an effect

Few full languages are
exclusive one of the other

Imperative: Assembly
So

pure functional part
arithmetic

Scala scripts

> scala script.scala

Scala interpreter

> scala
Make script into application

object myApp extends App {
  ... script ...
}

3

Scala is statically-typed

- check validity of operators before running the program
  = evaluating the expression

   "static" = before running program
   "dynamic" = while running program
Meta-variables

$e$ expression
$v$ value
$t$ type

\[ e \rightarrow e' \]

Expression $e$ takes one step to $e'$

$\left( 1+2 \right) + \left( 3+4 \right) \rightarrow 3 + \left( 3+4 \right)$

$e_A$

No

Yes

$\Rightarrow \text{left-to-right eval}$

$\left( 1+2 \right) + \left( 3+4 \right) \rightarrow 25 + \left( 3+4 \right)$
\[ e \rightarrow^* e' \] (0 or more)

\( e \) makes some number of steps to \( e' \)

\[(1+2) + (3+4) \rightarrow 7 \quad \rightarrow 4 \quad 3 + 7 \quad \rightarrow 7 \star \quad 3 + (3+4) \quad \rightarrow 10\]

\[ e \Downarrow \] \( \nu \) exp \( e \) evaluates to value \( \nu \)

\[(1+2) + (3+4) \Downarrow 10\]