



## Lab 1 Discussion

---

1. Get started - two weeks left
2. The lab manipulates the syntax of the input program, not its meaning or value
  - Read the code into some data structure
  - Rename values so that each operation assigns a unique name
    - (only valid in straightline code; makes names  $\equiv$  to live ranges)
  - Run the allocator
  - Write the results back out as valid ILOC
3. First step - write something to read and write the ILOC
  - Keep it simple
4. If you look at the example code in § 13.2 of EaC, keep in mind that you only have 1 register class in your machine
  - Substantial simplification to the code

## Lab 1 Discussion

---



### Reading and writing ILOC

- Use a simple data structure - say a  $k \times 4$  array of integers
  - Turn opcode into an integer
    - Simple function using string comparisons
  - Represent registers and immediates as integers
  - Prettyprinter can interpret those ints contextually
- This part of the lab should take no more than an evening (The allocators are the tricky part, not the I/O.)