Part 0
Preliminaries

What is a Computer?
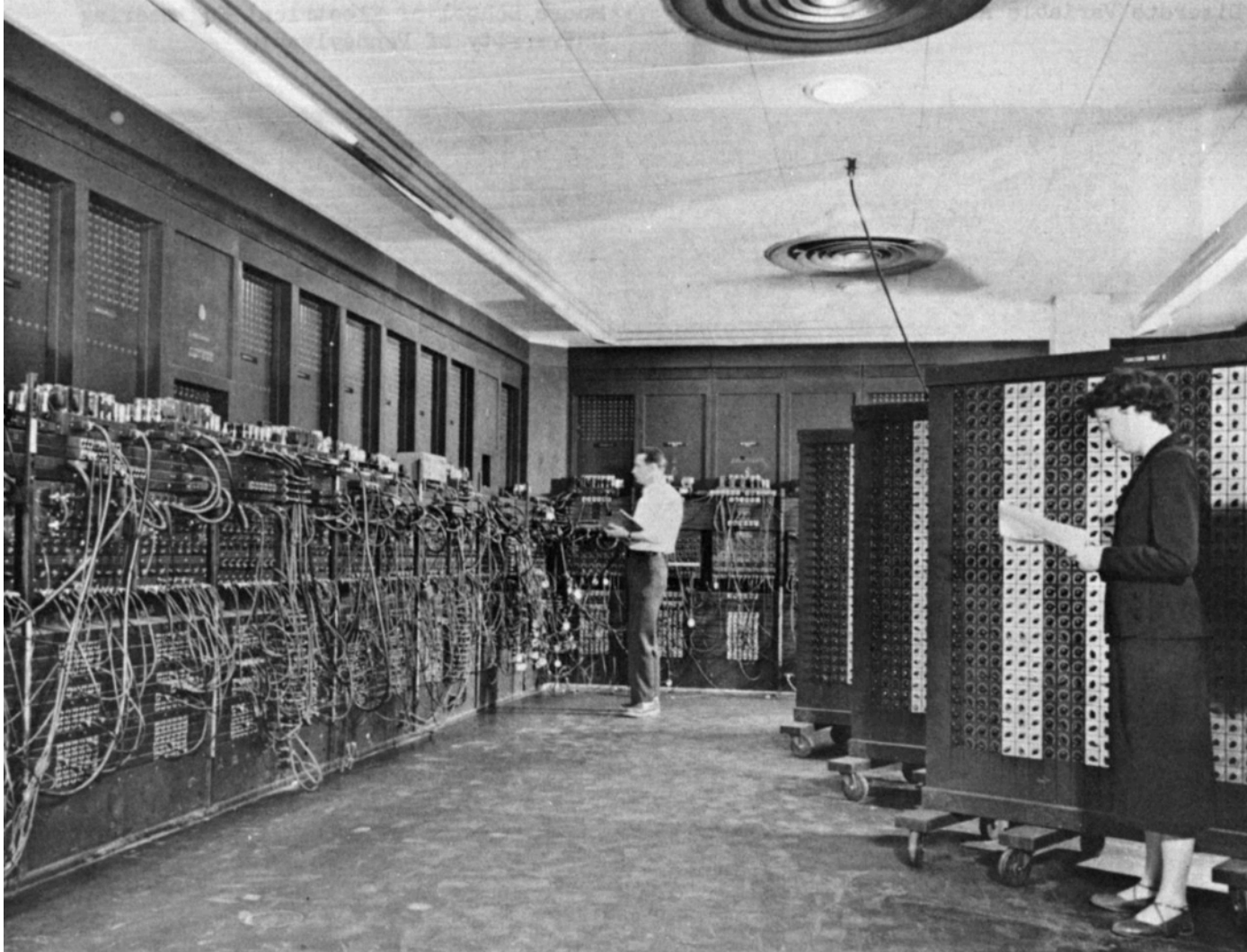
What is a Computer?

CPU (Central Processing Unit)

registers

ALU	math

R0
R1
R2
R3
R4
R5
R6
R7

control unit

PC
SP

load

store

memory

data

```
5548 89e5 897d
fc00 0f8e 1800
45f8 8b45 fc83
8b45 f85d c300
```

program

```
ADD R1, R2, R3
SUB R3, R4, R5
...
STO R5, (R6)
```
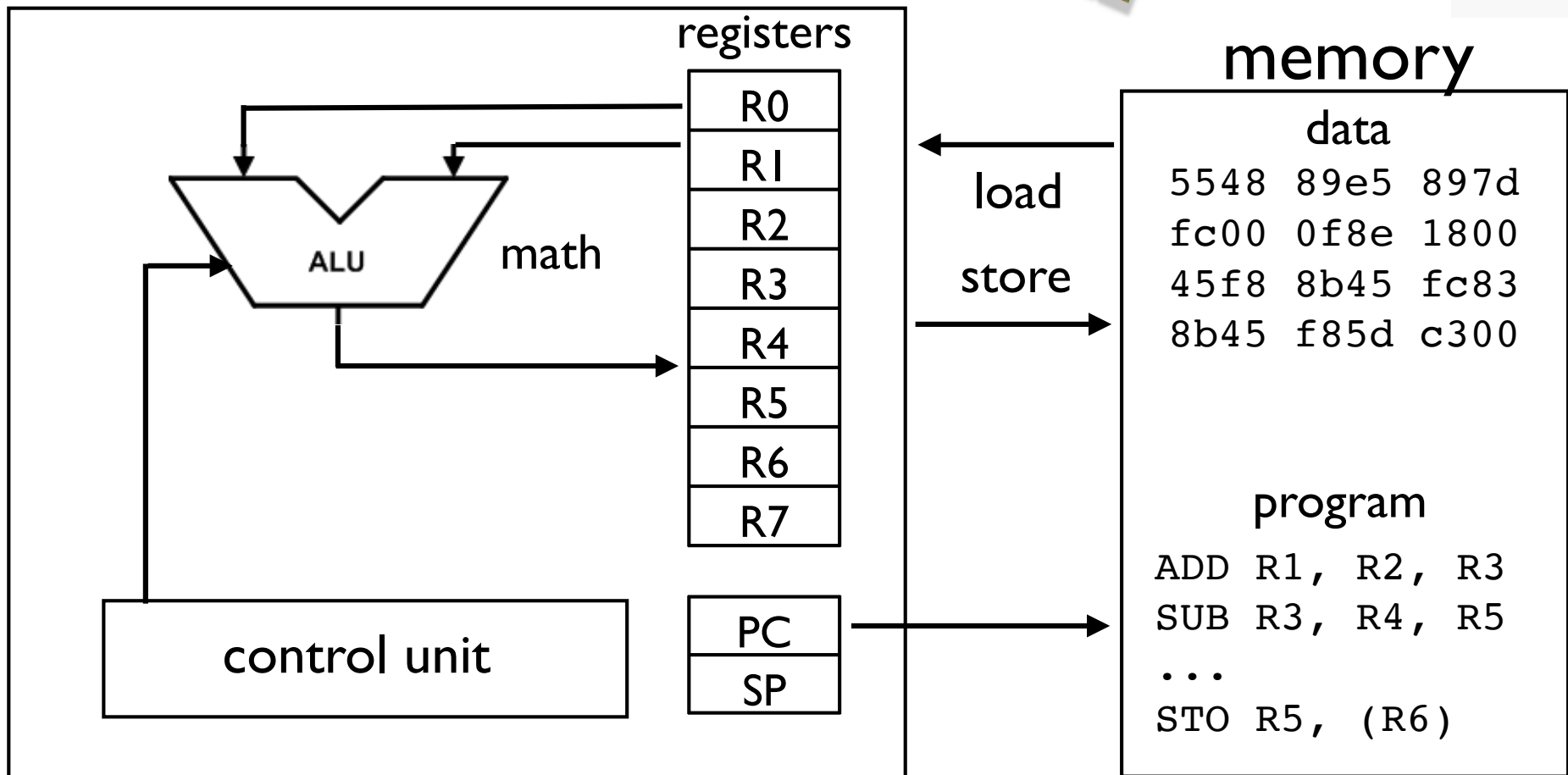
What is a Computer?

- Computers are not especially "smart"

- Glorified calculator (with program memory)

- Example:

```
2 + 3 * 4 - 5

MOV 2, R1              ; R1 = 2
MOV 3, R2              ; R2 = 3
MOV 4, R3              ; R3 = 4
MUL R2, R3, R4         ; R4 = R2 * R3 = 3 * 4 = 12
ADD R1, R4, R5         ; R5 = R1 + R4 = 2 + 12 = 14
MOV 5, R6              ; R6 = 5
SUB R5, R6, R7         ; R7 = R5 - R6 = 14 - 5 = 9
```
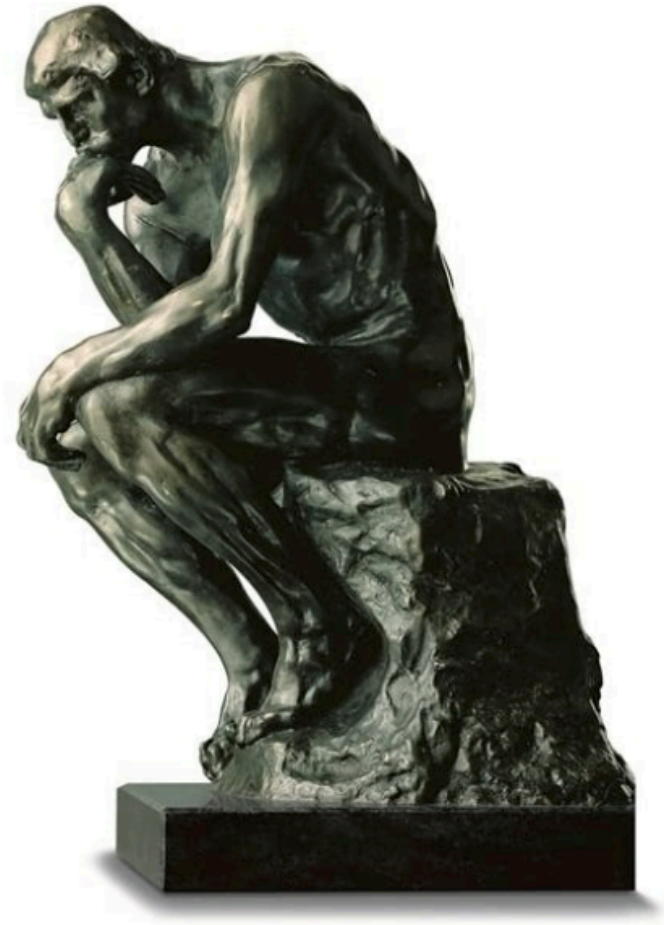
What is Computation?

"Computer Science is no more about computers than astronomy is about telescopes."

What does it actually mean to "compute" something?

What can be computed?

What is Computation?

- It's a process or procedure

```
Compute: 5!      -> 5 * 4 * 3 * 2 * 1

result = 1
n = 5
while n > 0:
    result = result * n
    n = n - 1
```

- Step-by-step sequence of operations

What is Computation?

- But, what is the actual essence of "computation?"

Deep Idea

- Perhaps computation is sequenced substitution

```
2  +  3  *  4  –  5
                  │ substitute (eval 3 * 4)
                  ▼
      2  +  12  –  5
              │ substitute (eval 2 + 12)
              ▼
        14  –  5
            │ substitute (eval 14 - 5)
            ▼
            9
```

- You substitute one thing for another after performing <u>single</u> steps in a certain order

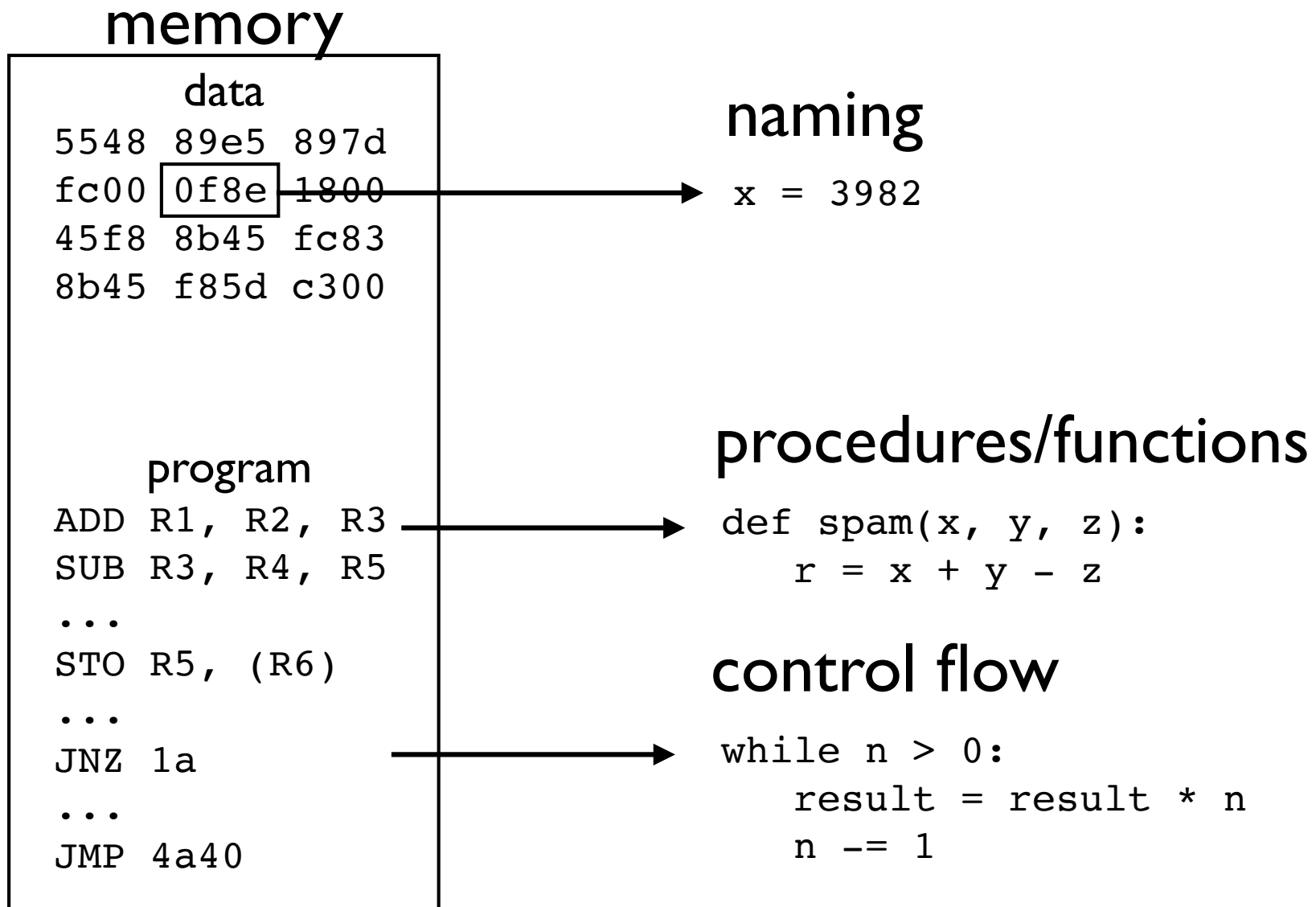- Stop when no more steps are possible

Substitution

- Substitution is a major facet of compiler writing

- "High level" things are replaced by "lower level" things (repeat until you can't go any further)

- Part of moving from an abstract programming language down to the real world

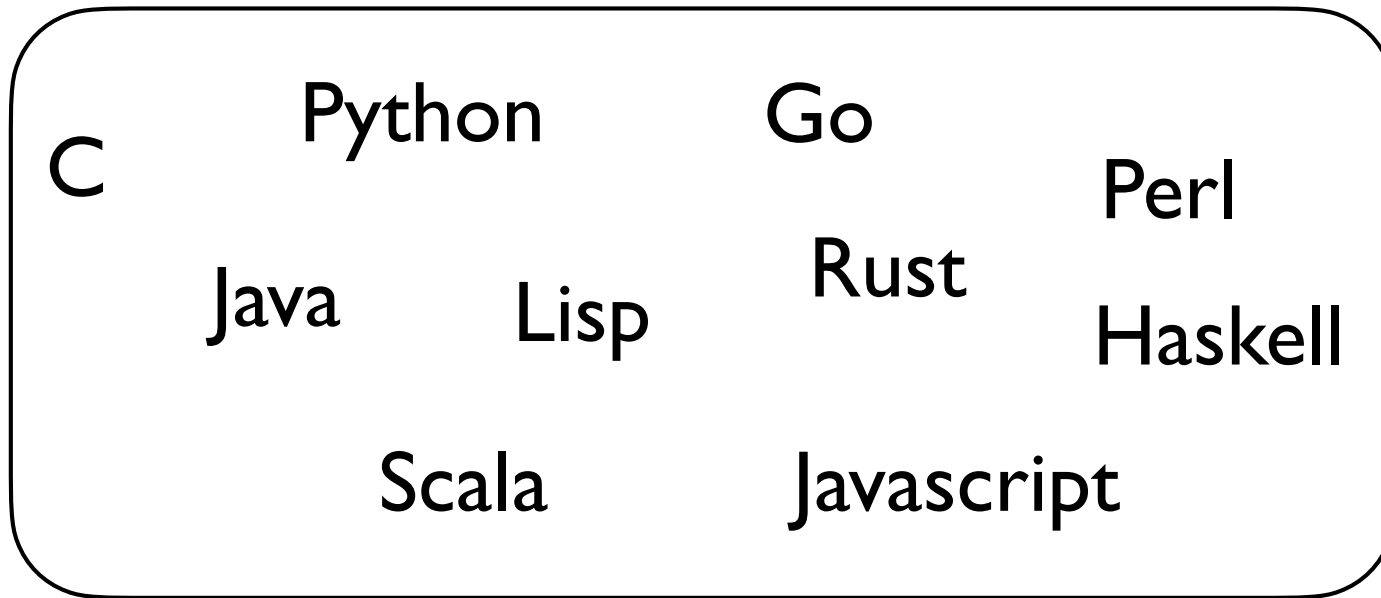- Also a major part of the mathematical foundations of programming languages.

What is Programming?

- Describes a computational process. Yes.

- How?

- Key feature: <u>Abstractions</u>

Programming is Abstraction

memory

```
        data
5548  89e5  897d
fc00 │0f8e│ 1800
45f8  8b45  fc83
8b45  f85d  c300
```

naming

```
x = 3982
```

```
      program
ADD R1, R2, R3
SUB R3, R4, R5
...
STO R5, (R6)
...
JNZ 1a
...
JMP 4a40
```

procedures/functions

```
def spam(x, y, z):
    r = x + y - z
```

control flow

```
while n > 0:
    result = result * n
    n -= 1
```

Programming is Abstraction

C   Python   Go   Perl
    Java   Lisp   Rust   Haskell
    Scala   Javascript

Abstraction

"Metal"

010111111   00001111
110001111   11011111
011111011   Bits   11111111
110111111100000000000000000000000
110000000000000100000000000000000

Big Picture

Computers

Computation

You
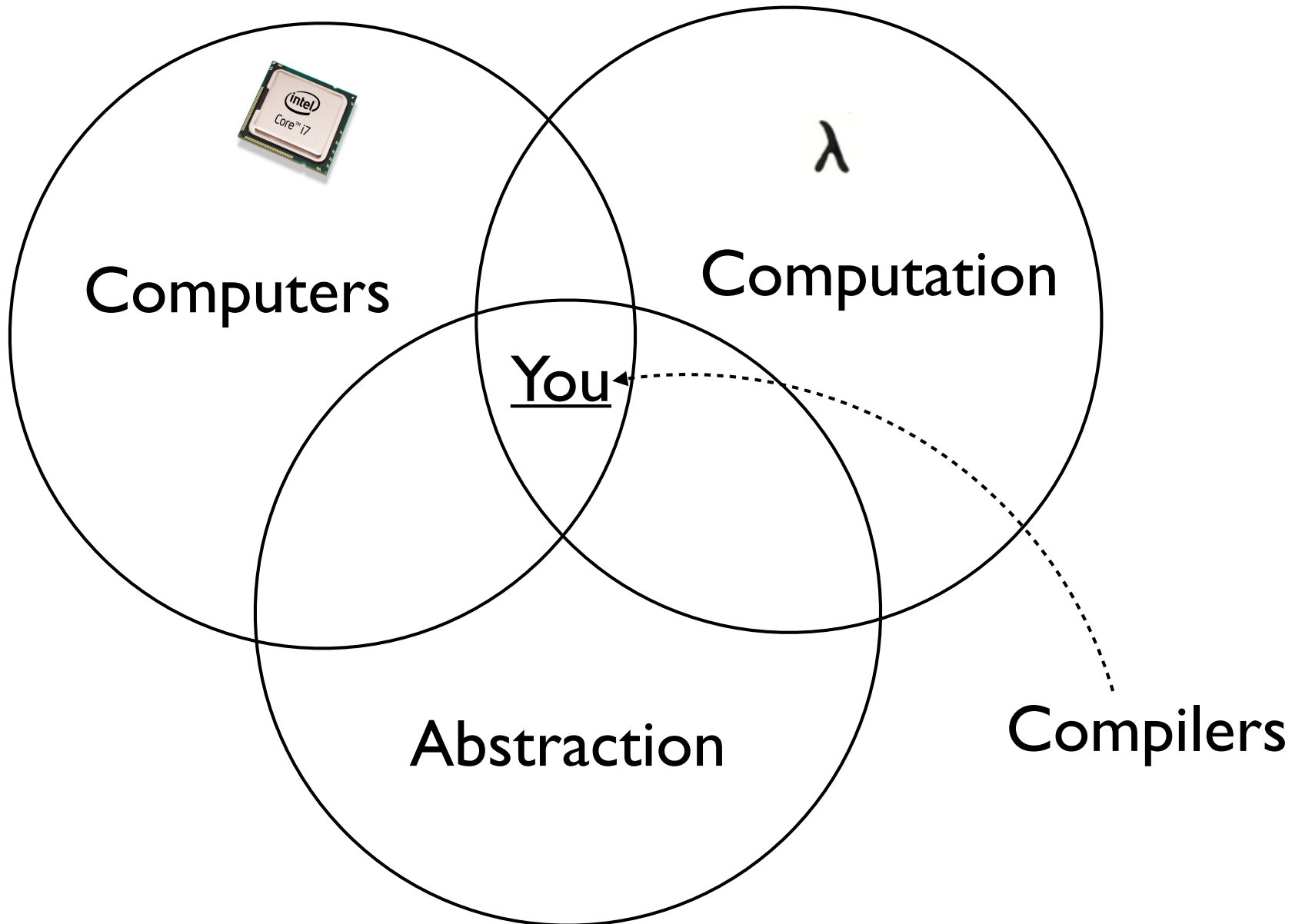
Abstraction

Compilers

Project 0

Find the file `metal.py`

Follow the instructions found inside.

Project 0.5

Play with wabbit

Find the directory `SillyWabbit/`

Follow instructions in the README