“I’m not a computer”:
How identity informs value and expectancy during a programming activity

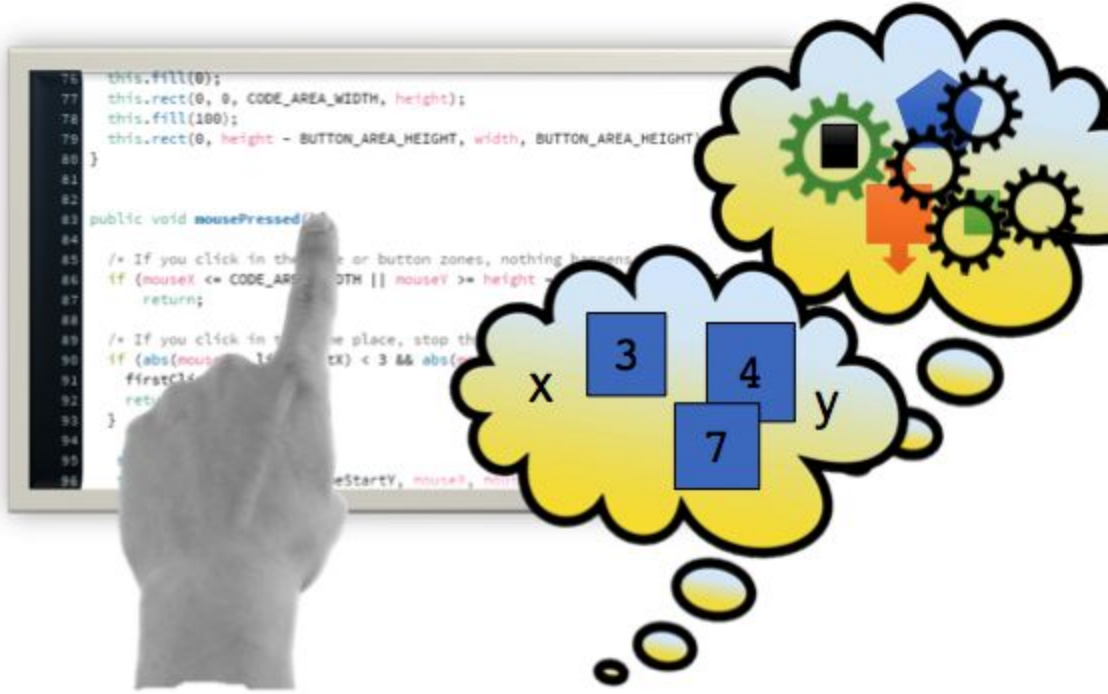
Kathryn Cunningham[1], Rahul Agrawal Bejarano[2],
Mark Guzdial[2], Barbara Ericson[1]

University of Michigan
[1]School of Information [2]Computer Science and Engineering

Code tracing is mentally simulating code execution.

Involves careful tracking of variables and control flow.

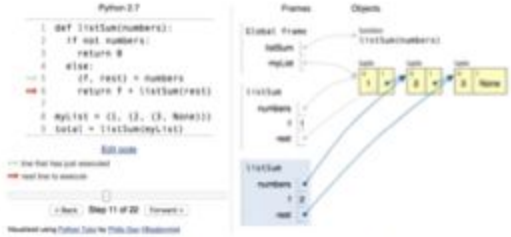
Tracing problems and visual tools help learners build code tracing skills.



What are the final values of x and y?

```
x = 3
y = 4
x = y
y = 7
```

Tracing problems



Python Tutor (Guo)

UUhistle (Sorva)

Hierarchies of programming skills for an undergraduate setting typically place **code tracing early**, as something students naturally learn first (Lister 2011), or should learn first for deeper understanding (Xie et al. 2019).

1. Tracing code
2. Explaining code "in plain english"
3. Writing code

Lister 2011

1. Tracing code (Reading semantics)
2. Writing semantics
3. Reading templates
4. Writing templates

Xie et al. 2019

Problems that teach & test programming skills are often **stripped of larger context**, so learners are not "distracted" or "influenced" by clues like meaningful variable names (Lister et al. 2004).

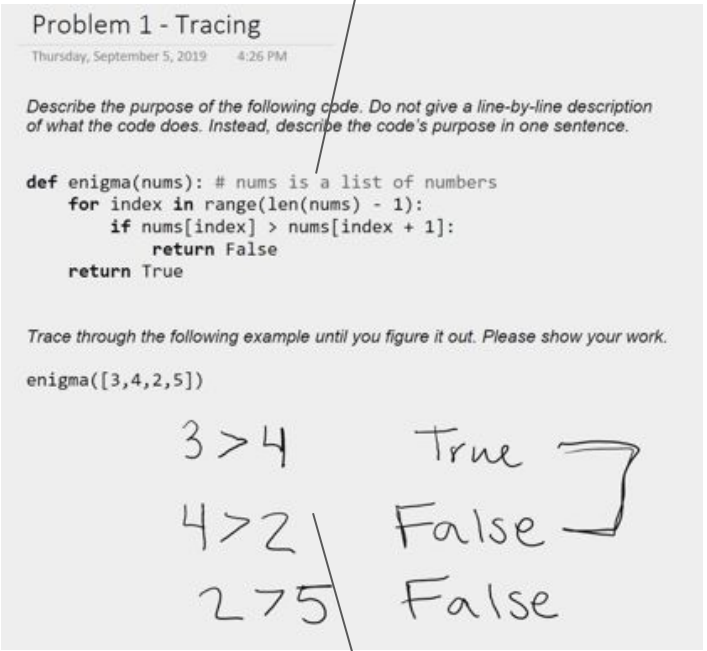
```
def enigma(nums): # nums is a list of numbers
    for index in range(len(nums) - 1):
        if nums[index] > nums[index + 1]:
            return False
    return True
```

Lister, Fidge, & Teague 2009

Study design

Originally: a cognitive analysis of the ways code tracing helps novice programmers explain the purpose of code in natural language (Lopez, Whalley, Robbins, & Lister, 2008).

Problems from past studies



A participant's trace

Study design

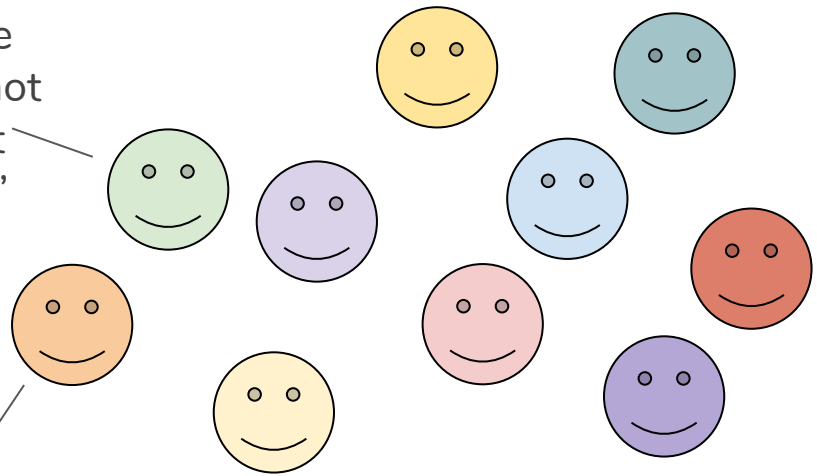
Originally: a cognitive analysis of the ways code tracing helps novice programmers explain the purpose of code in natural language (Lopez, Whalley, Robbins, & Lister, 2008).

But!

***Affective responses* to code tracing were unexpected, common, and varied!** For some learners, code tracing is not only challenging, but seems to have some “baggage”…

12 participants

- 1-2 formal college-level programming courses
- 8 undergraduates, 4 graduate students
- 8 women, 4 men
- Majority from an Information major



"I'm like, objective oriented. So it's not necessarily about each line of code"

"In [my course], they definitely had us draw a grid [to trace], and I definitely didn't do that"

"You don't do much coding by hand, obviously"

"This is bad but I very infrequently count through the iterations"

"I should know this"

Problem 2b - Tracing

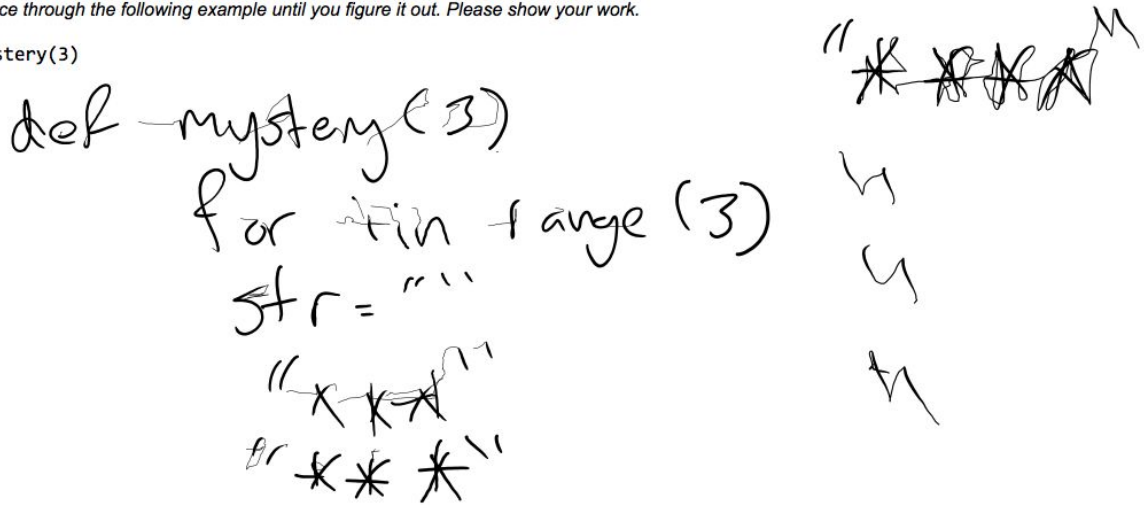
Thursday, September 5, 2019 4:26 PM

Describe the purpose of the following code. Do not give a line-by-line description of what the code does. Instead, describe the code's purpose in one sentence.

```
def mystery(num):
    for i in range(num):
        str = ""
        for i in range(num):
            str += "*"
        print(str)
```

Trace through the following example until you figure it out. Please show your work.

```
mystery(3)
```



Charles: *The inner for loop? Oh, I don't know. It's kinda confusing to me.*

Interviewer: *Yeah?*

Charles: *Um, I guess like, in general, I found that like, the least helpful thing in my programming class was reading code, as weird as that sounds.*

Problem 1 - Tracing

Thursday, September 5, 2019 4:26 PM

Describe the purpose of the following code. Do not give a line-by-line description of what the code does. Instead, describe the code's purpose in one sentence.

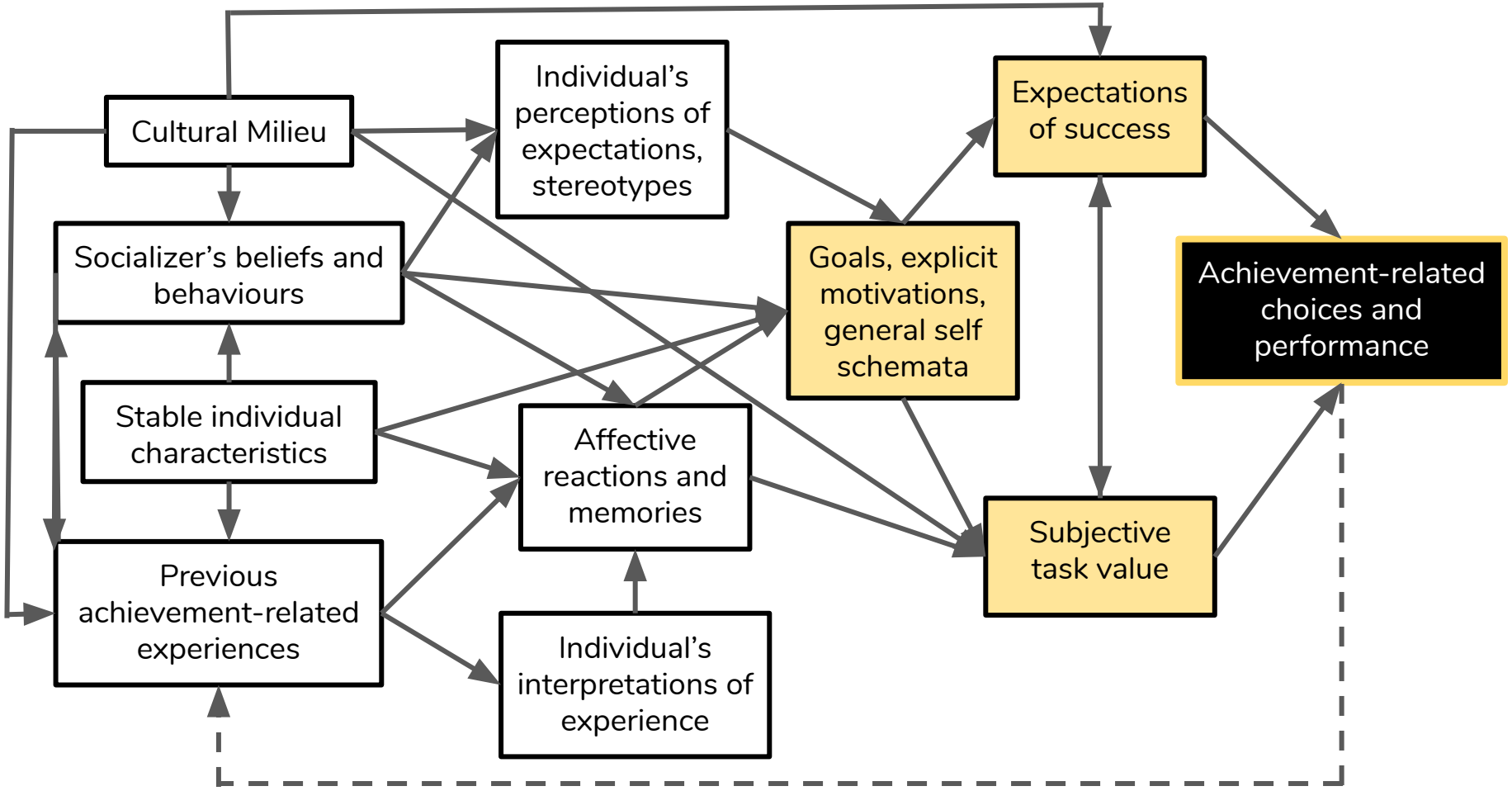
```
def enigma(nums): # nums is a list of numbers
    for index in range(len(nums) - 1):
        if nums[index] > nums[index + 1]:
            return False
    return True
```

Trace through the following example until you figure it out. Please show your work.

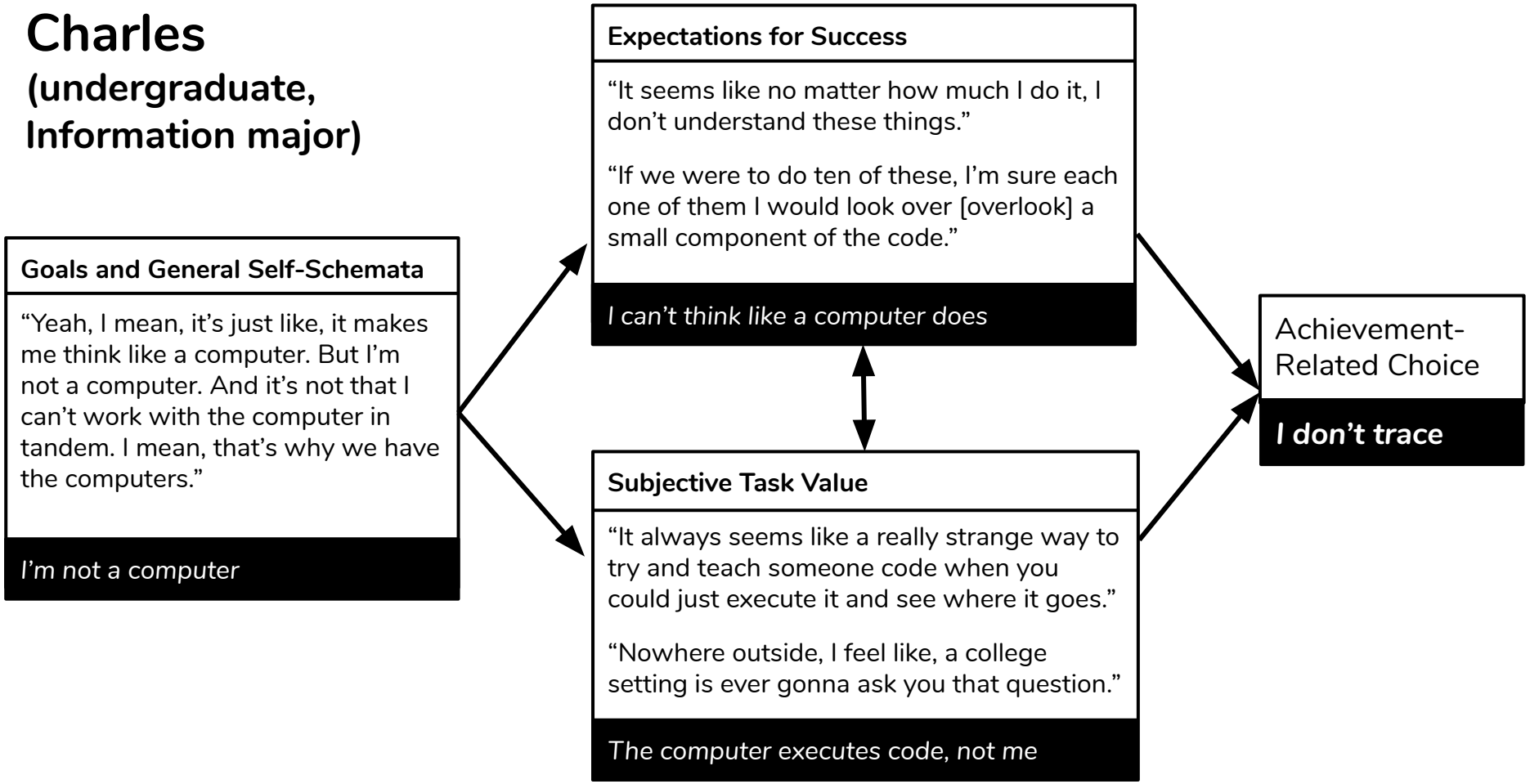
```
enigma([3,4,2,5])
```

Luke: "*I hated this sort of work honestly.*"

Eccles Expectancy-Value Model of Achievement Choice



Charles
(undergraduate, Information major)



Charles
(undergraduate, Information major)

Goals and General Self-Schemata

"Yeah, I mean, it's just like, it makes me think like a computer. But I'm not a computer. And it's not that I can't work with the computer in tandem. I mean, that's why we have the computers."

I'm not a computer

Expectations for Success

"It seems like no matter how much I do it, I don't understand these things."

"If we were to do ten of these, I'm sure each one of them I would look over [overlook] a small component of the code."

I can't think like a computer does

Subjective Task Value

"It always seems like a really strange way to try and teach someone code when you could just execute it and see where it goes."

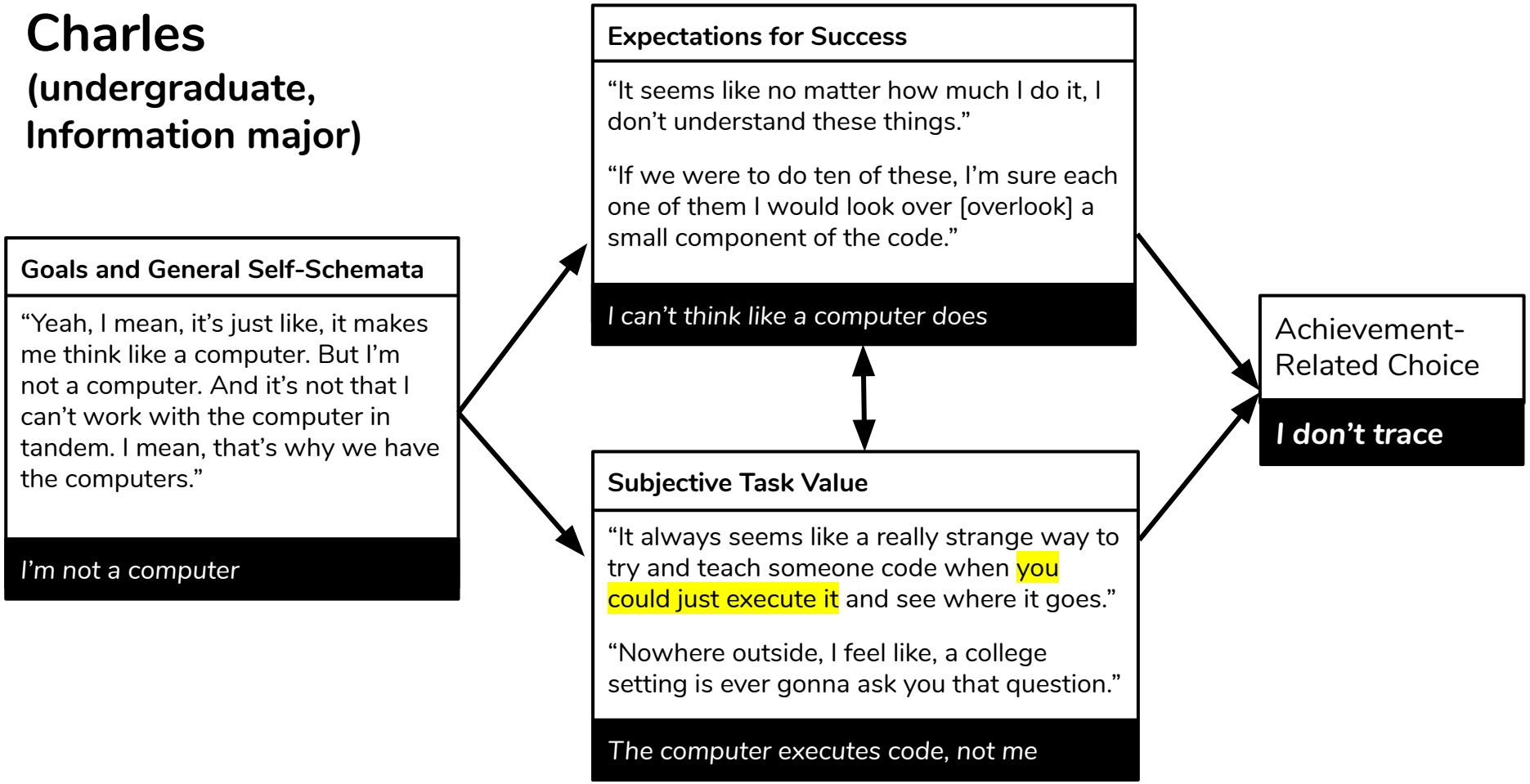
"Nowhere outside, I feel like, a college setting is ever gonna ask you that question."

The computer executes code, not me

Achievement-Related Choice

I don't trace

Charles
(undergraduate, Information major)



Charles
(undergraduate, Information major)

Goals and General Self-Schemata

"Yeah, I mean, it's just like, it makes me think like a computer. But I'm not a computer. And it's not that I can't work with the computer in tandem. I mean, that's why we have the computers."

I'm not a computer

Expectations for Success

"It seems like no matter how much I do it, I don't understand these things."

"If we were to do ten of these, I'm sure each one of them I would look over [overlook] a small component of the code."

I can't think like a computer does

Subjective Task Value

"It always seems like a really strange way to try and teach someone code when you could just execute it and see where it goes."

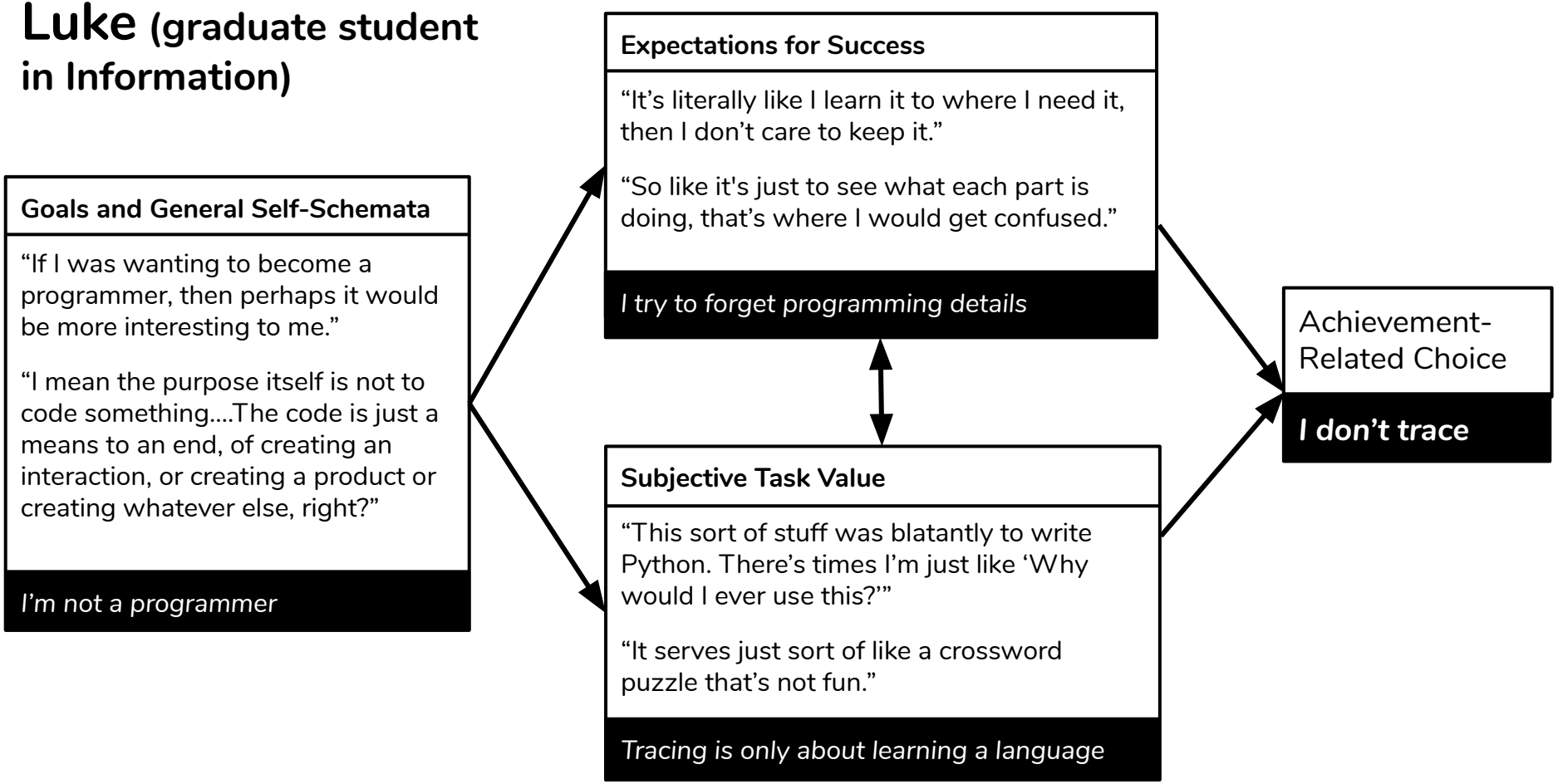
"Nowhere outside, I feel like, a college setting is ever gonna ask you that question."

The computer executes code, not me

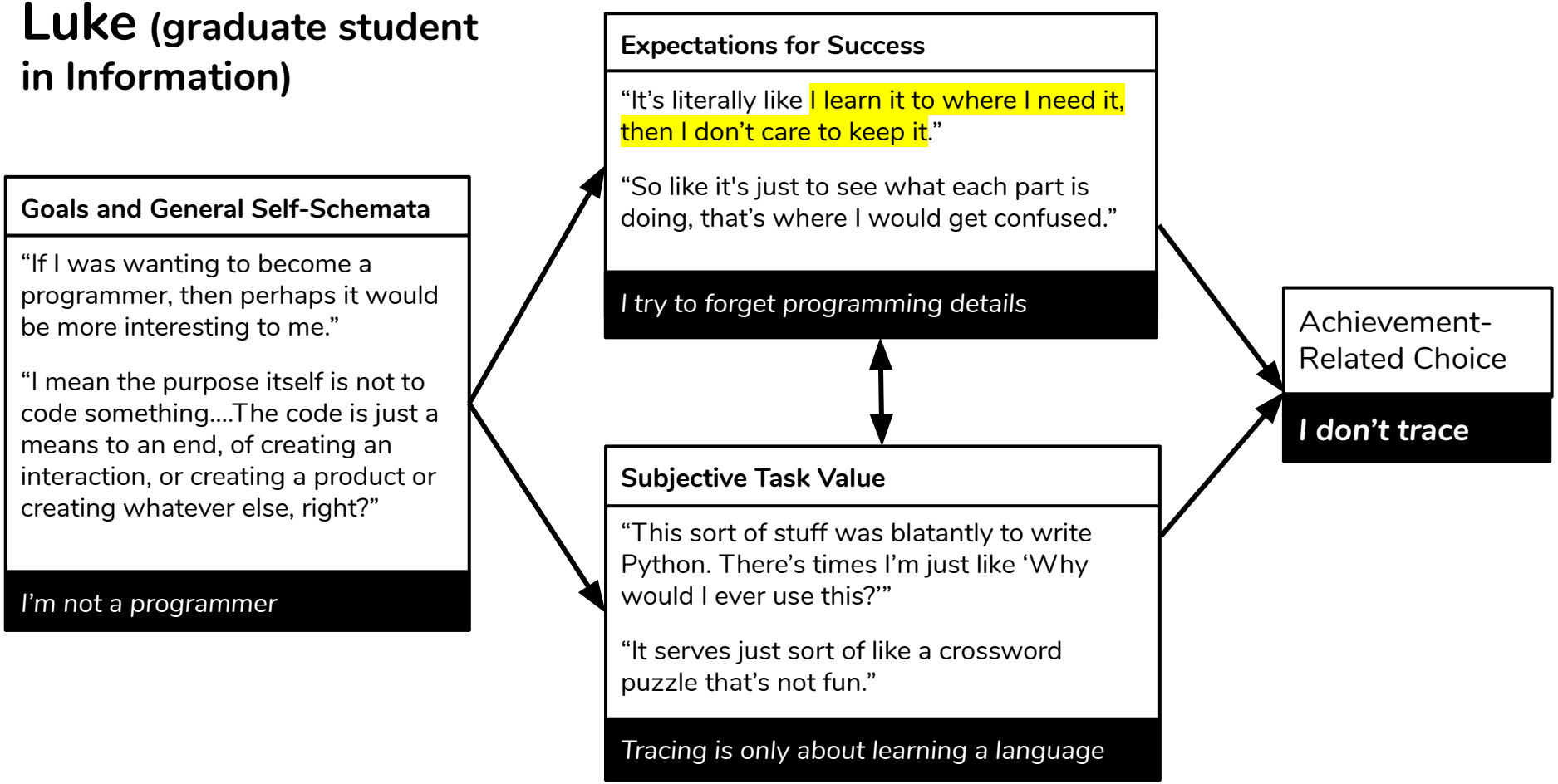
Achievement-Related Choice

I don't trace

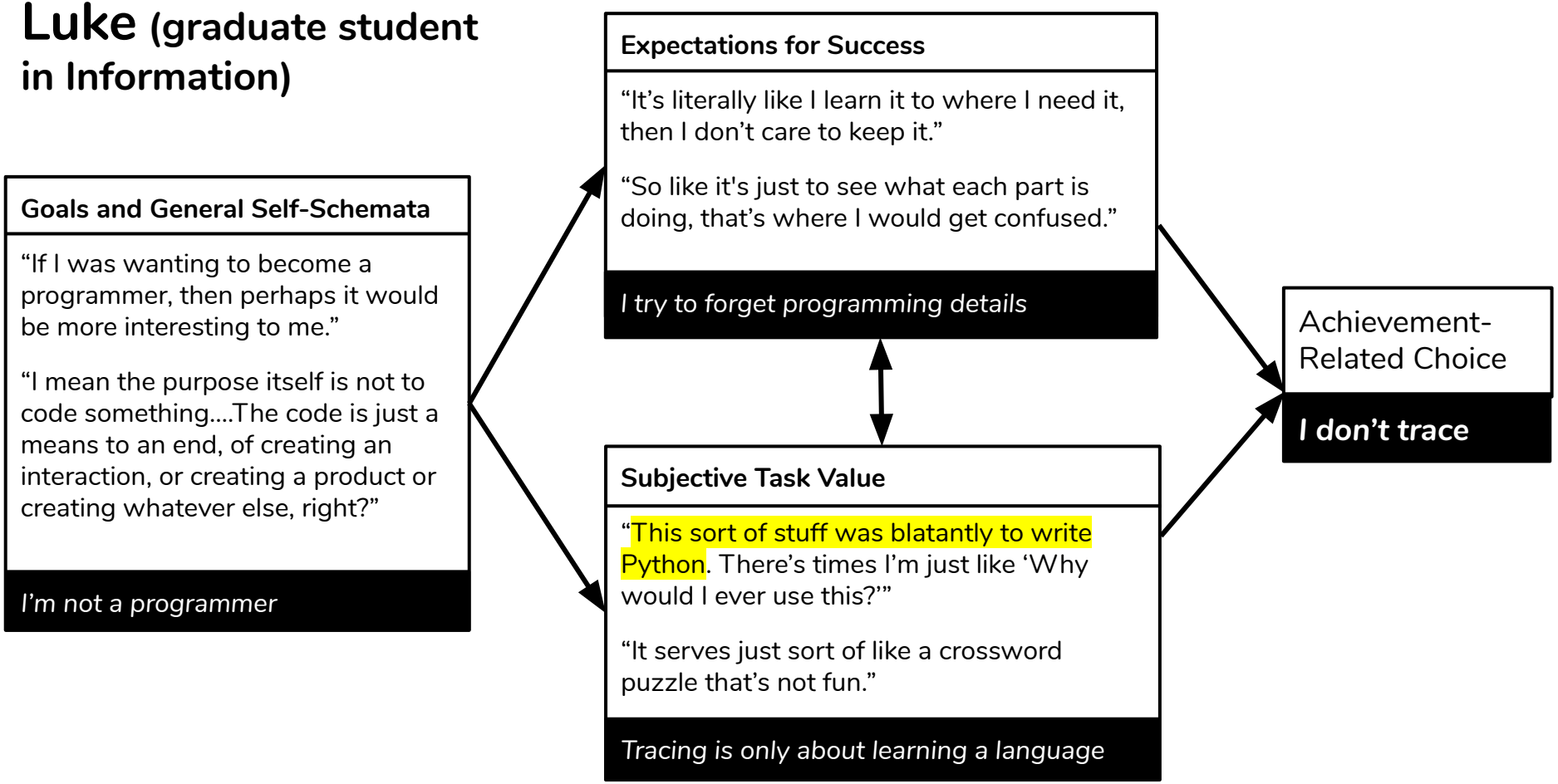
Luke (graduate student in Information)



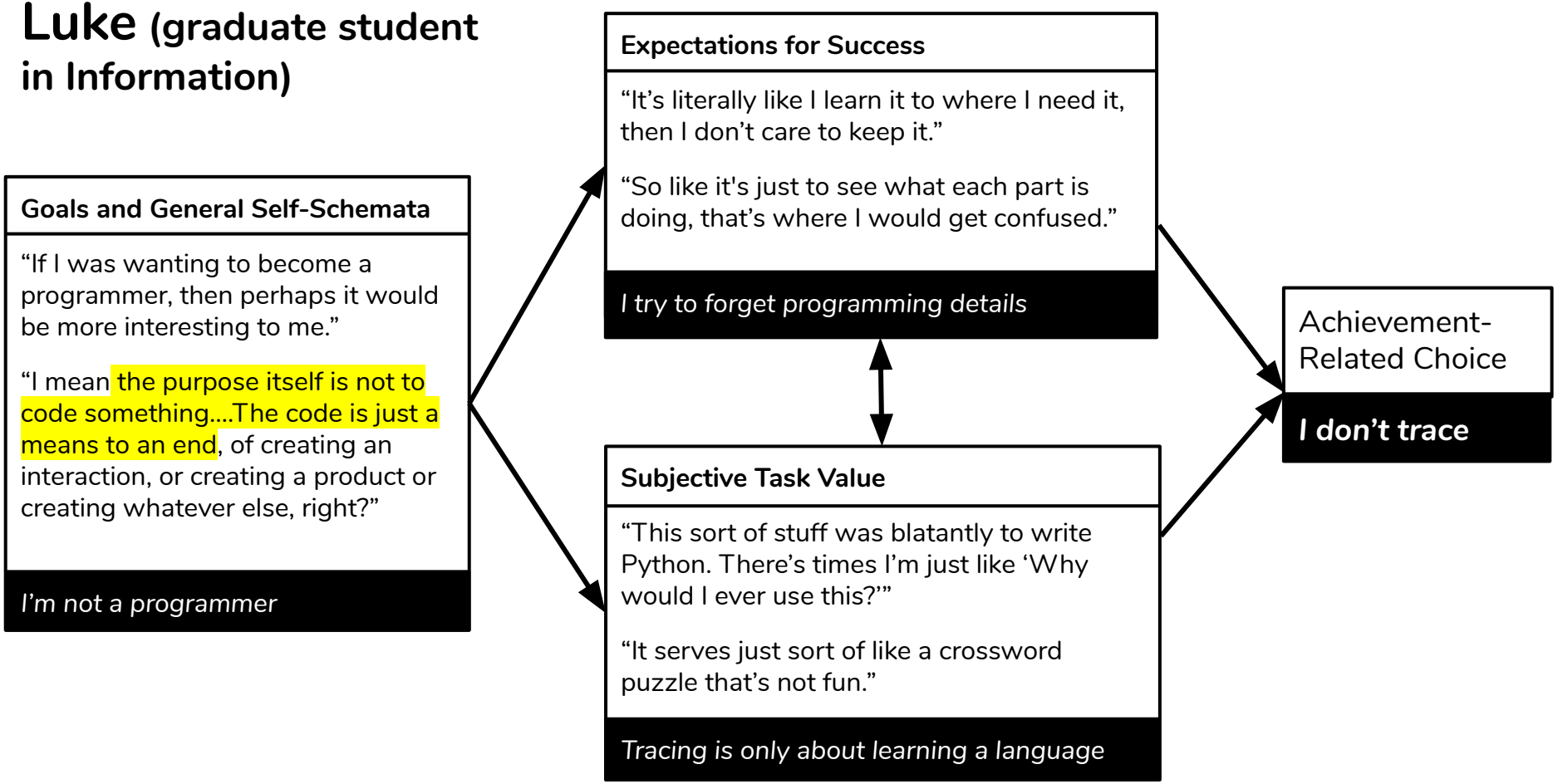
Luke (graduate student in Information)



Luke (graduate student in Information)



Luke (graduate student in Information)



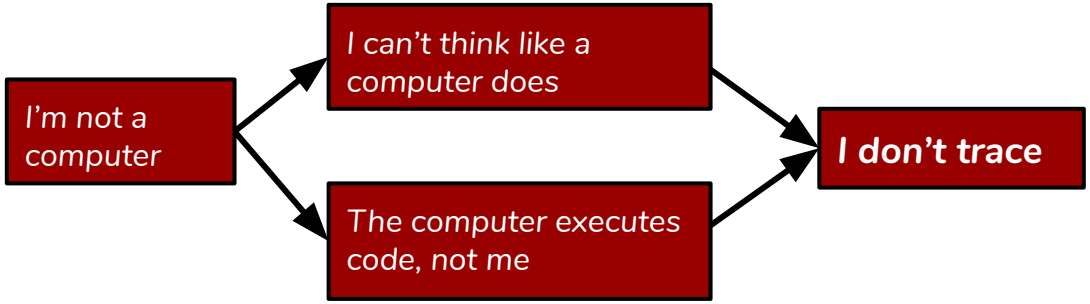
Takeaways

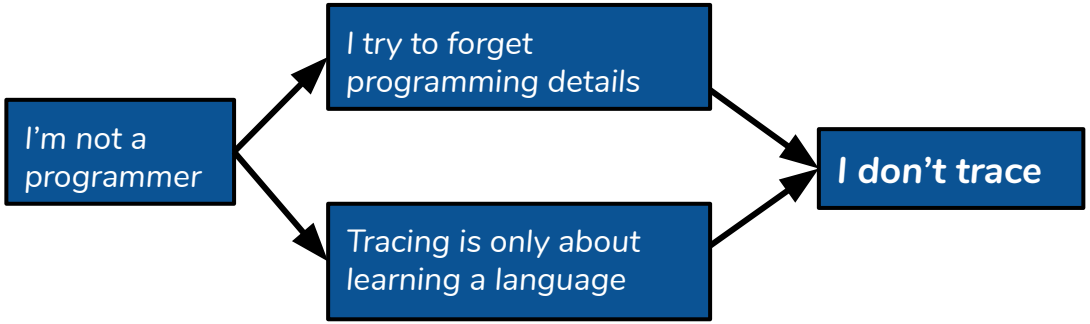
Programming learning pathways should consider affective and identity factors.

Maybe we shouldn't require tracing early for all learners. Non-CS majors in particular could benefit from an alternative pathway to build expertise in programming.

This alternative pathway should be:

- Function-oriented,
- Contextualized,
- Authentic





Thanks!!

@katieirenec
kicunn@umich.edu