YOUR TURN

- Context
- Materials
- Task format
- Data collection
- Evaluation

- Design one task for a usability study of your language feature.
- For now, assume you have a full implementation.
- Or: add pairs to Java. Vague hypothesis: more "efficient" to have a uniform way of doing pairs than making people use little classes all the time.

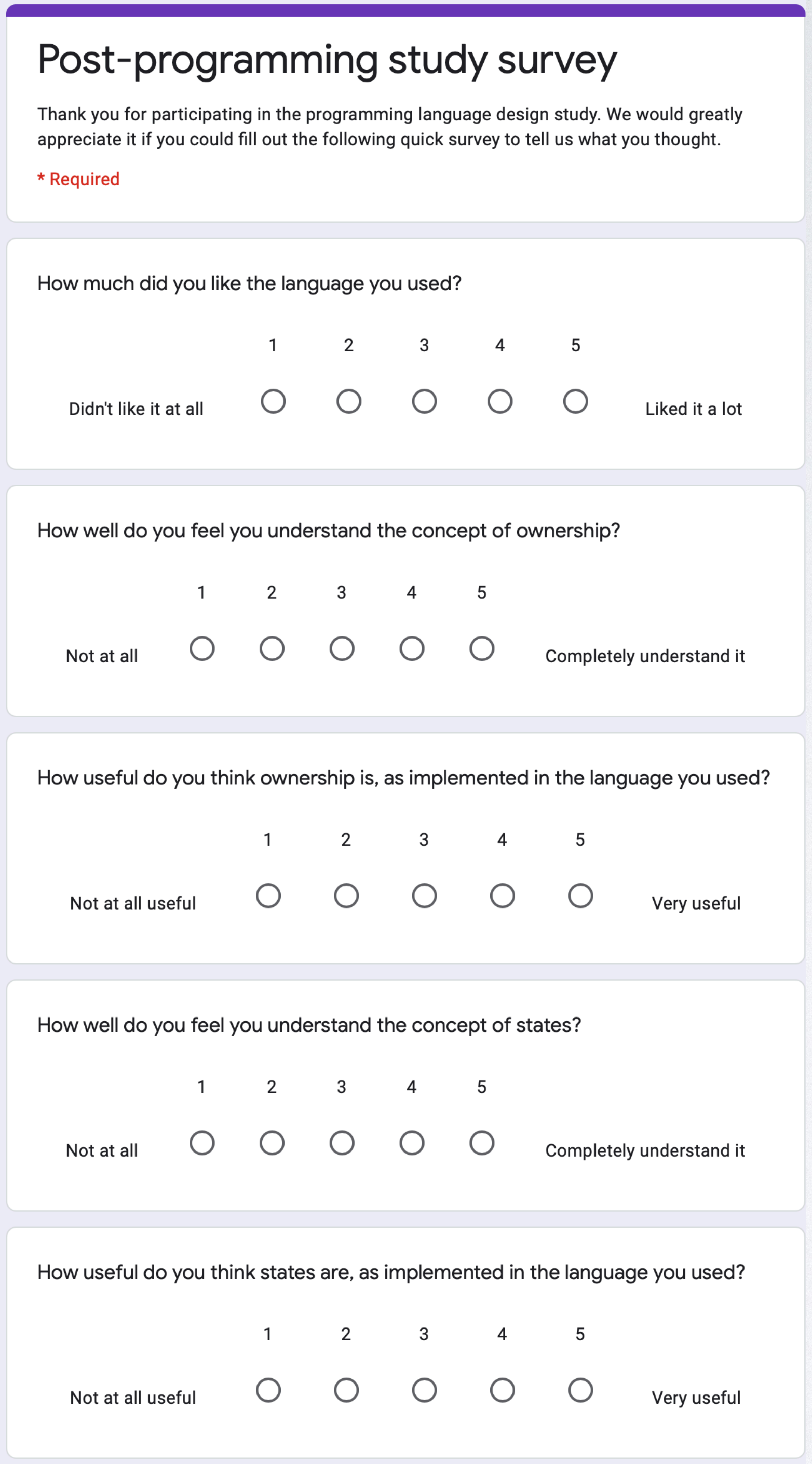
```
(int, int) intPair = (1, 2);

println(intPair._1); // 1

println(intPair._2); // 2
```

POST-STUDY SURVEY

- Collect quotes you can use
- Did people like it?



Post-programming study survey

Thank you for participating in the programming language design study. We would greatly appreciate it if you could fill out the following quick survey to tell us what you thought.

* Required

How much did you like the language you used?

	1	2	3	4	5	
Didn't like it at all	○	○	○	○	○	Liked it a lot

How well do you feel you understand the concept of ownership?

	1	2	3	4	5	
Not at all	○	○	○	○	○	Completely understand it

How useful do you think ownership is, as implemented in the language you used?

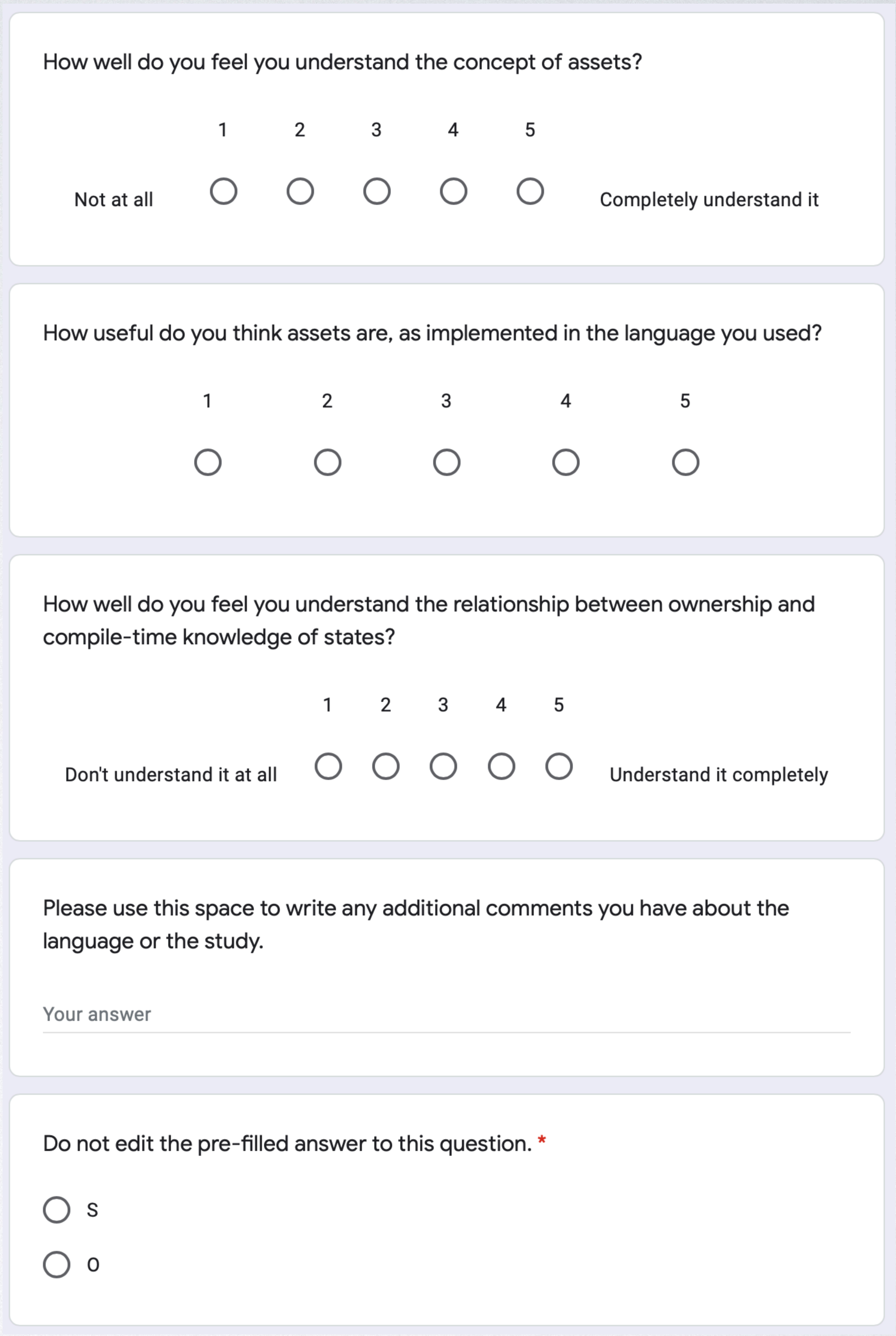
	1	2	3	4	5	
Not at all useful	○	○	○	○	○	Very useful

How well do you feel you understand the concept of states?

	1	2	3	4	5	
Not at all	○	○	○	○	○	Completely understand it

How useful do you think states are, as implemented in the language you used?

	1	2	3	4	5	
Not at all useful	○	○	○	○	○	Very useful



How well do you feel you understand the concept of assets?

	1	2	3	4	5	
Not at all	○	○	○	○	○	Completely understand it

How useful do you think assets are, as implemented in the language you used?

1	2	3	4	5
○	○	○	○	○

How well do you feel you understand the relationship between ownership and compile-time knowledge of states?

	1	2	3	4	5	
Don't understand it at all	○	○	○	○	○	Understand it completely

Please use this space to write any additional comments you have about the language or the study.

Your answer

Do not edit the pre-filled answer to this question. *

- ○ S
- ○ O

HOW MANY?

- Until it's not useful anymore.

SCAFFOLDING

- I only have a language *prototype*. How do I evaluate my language?

WIZARD OF OZ

- Have an experimenter simulate incomplete parts of the system
- Can give any kind of programming task this way if it doesn't require running the code.

NATURAL PROGRAMMING

- Want to find out how people "naturally" specify software.
- Yes, depends on prior experience
    - Find people without experience
    - Put people in a novel situation
- Constrain choices

EXAMPLE

- Suppose we want to design a system that allows users to create and participate in lotteries. The rules of the lottery are as follows:

- Someone creates a lottery with a secret "winning number" between 0 and 100.

- Anyone can purchase a lottery ticket for a set amount of money. A lottery ticket also has a number (picked by the buyer) between 0 and 100. If the lottery ticket's number is equal to the lottery's number, that is a winning ticket.

- The buyer of a ticket can check whether a ticket is the winning ticket and redeem a winning lottery ticket for a set amount of money. The buyer of a ticket can redeem the ticket at any point after they buy it.

- It is only possible to redeem a lottery ticket from the lottery where the ticket was purchased. For instance, if you buy a lottery ticket from lottery1, you cannot redeem your ticket from lottery2, even if that ticket's number matches the winning number of lottery2.

EXAMPLE (CONT.)

- Design, using pseudocode, a program to handle this lottery system. Do not worry about writing code that resembles any particular language, and feel free to make up any language features you may want. We want to see the kind of code you would want to be able to write. The goal here is to see how people naturally go about solving a problem like this.

- It is critical to the system that a lottery ticket can be redeemed only from the lottery where it was purchased. You can assume that creator and players of the lottery have accounts of some sort from which money can be withdrawn and deposited.

BACK-PORTING DESIGN CHOICES

- Context: designing a feature in a new language. Want to:
  - Analyze choice A
  - Compare choices A and B
- Teach people language L + A, L + B?
  - People get confused: was it L, or A?
- Alternative: add A to a *standard language*.

YOUR TURN

- Design an early-stage usability study using one of the above techniques.

QUALITATIVE STUDIES

- Data sources:
  - In-depth, open-ended interviews
  - Direct observation
  - Written documents
- "Research, like diplomacy, is the art of the possible." (Patton)

ETHNOGRAPHY

- Immerse yourself in the environment.
- Long-form results (e.g. a book)
- Example: "This is what I learned from a year living with the X people."

INTERVIEWS

- Prepare questions in advance
- "Semi-structured" — not restricted to your script
- Ask about experiences

PREPARING AN INTERVIEW

- Write questions in advance
  - Including probes (follow-ups)
- Ask for permission to record (PA is a two-party consent state)
- Phrase questions neutrally
- Progress from general to specific

Specific probes, but also:
"How do you mean that?"
"Tell me more about that."
"Anything else?"

1. How long have you been programming professionally?
2. Can you give an order of magnitude estimate of the size of the largest project you've made significant contributions on? Number of people, lines of code?
3. In what programming languages do you consider yourself proficient?
4. How did you get into software development? Do you have a computer science background?
5. Let's talk about changes that happen to state in software you've worked on. Many kinds of software maintain state, such as object graphs, files, or databases, but there's a possibility of corruption during changes due to bugs. How do you make sure that state in running programs remains valid?
    1. Are there specific techniques do you use? If so, what are they?
    2. Do you sometimes want to make sure that some operations don't change any state or don't change certain state?
        1. Tell me about a recent time you did this.
        2. How often does this come up?
        3. Do you use language features to help?
    3. Do you sometimes want to make sure that some state never changes?
        1. Tell me about a recent time you did this.
        2. How often does this come up?
        3. Do you use language features to help?
    4. Do you sometimes want to make certain kinds of modifications to state impossible for some users of an API but not others? If so, how do you do that?
6. How often do you work on concurrent aspects of software? What mechanisms do you use to control concurrency?
    1. Do you use immutability to help address or prevent concurrency issues?
7. How much work have you done on security-related aspects of your software? Have you found or fixed any vulnerabilities?
    1. Do you use immutability to help address or prevent security issues?
8. Can you recall a recent situation in which you created an immutable class or other data? If so, tell me about it.
9. Can you recall a recent situation where you changed a class from immutable to mutable? If so, tell me about it.
10. Can you recall a recent situation in which you changed a class from mutable to immutable. If so, tell me about it.
11. Can you think of a bug you investigated or fixed that was caused by a data structure changing when it should not have? What was the problem and how did you solve it (if you solved it)?
    1. Would const have prevented the bug?
12. Have you ever tried using an immutable class and had it not work? Why not?
13. When you create a new class, how do you decide whether instances should be immutable?
14. Have you ever been in a situation where you wanted to use const or final but it didn't say what you wanted to say?
    1. or where you discovered you couldn't use it? What was the situation and why couldn't you use it?
15. Have you been in a situation where you had to revise your plan because something you'd assumed could mutate state was disallowed from doing so due to const?
16. Have you been involved in training new members of the team? What do you tell new members about immutability or ensuring invariants are maintained?
17. Sometimes, though an object is mutable after creation, it only needs to be changed for a short amount of time. For example, when creating a circular data structure, the cycle must be created after allocating all the elements. After that, however, the data structure doesn't need to be changed. Have you encountered situations like that? Do you think it would help if you could lock the object after all necessary changes were made?
18. Now, I'd like to move on to API design in general. Think of a recent API you designed.
    1. Did you make any conscious design or implementation decisions, to make the API easier or more manageable for these users?
    2. Are there any recurring issues / challenges that users have had with your API? How did you handle those?
    3. How do you differentiate between users of your API? Are there parts of the API that you expose some users but not others? How do you manage that?
    4. Did you make any conscious design or implementation decisions to protect key data or data structures from modification (inadvertent or malicious) from your users?

WRITE ONE INTERVIEW QUESTION

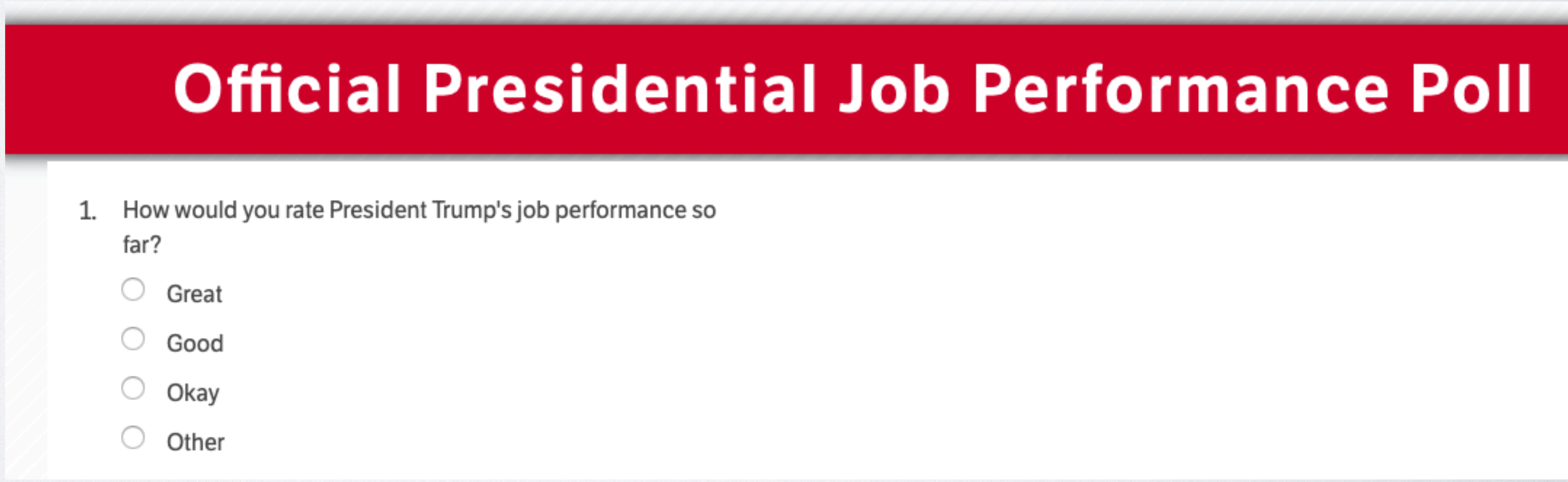
- Let's study debugging.

SURVEYS

- Used to generalize findings
- Select a population; then figure out how to sample from it
  - Incentives?
- Don't need a lot of responses for a *qualitative* study

QUESTION DESIGN

- Neutral phrasing
  - "C++ is better than Java, isn't it?"
- Balanced choices



Official Presidential Job Performance Poll

1. How would you rate President Trump's job performance so far?
   - Great
   - Good
   - Okay
   - Other