# How to create and use formative assessments at scale

Kristin Stephens-Martinez
Assistant Professor of the Practice
Duke University

# Outline

- Why should you listen to me? (a.k.a. My Background)
- What is a formative assessment?
- How do I use formative assessments?
- How do I write formative assessments?
- Q / A

Summary on blog: http://ksm-cs.blogspot.com/

# Why should you listen to me? (a.k.a. My Background)

- Dissertation: "Serving CS Formative Feedback on Assessments Using Simple and Practical Teacher-Bootstrapped Error Models"
- Analyze student wrong answers from code-tracing, constructed response, answer-until-correct questions

Reanswer exact same question until right

Read code to predict output

Create the answer

# Why should you listen to me? (a.k.a. My Background)

- Dissertation: "Serving CS Formative Feedback on Assessments Using Simple and Practical Teacher-Bootstrapped Error Models"
- Analyze student wrong answers from code-tracing, constructed response, answer-until-correct questions

Reanswer exact same question until right

Read code to predict output

Create the answer

- Learned a lot about how students get it wrong

# What is a formative assessment?

Formative                                                                                    Summative

- While learning is happening                    - After learning is done
- Low-stakes                                                   - High-stakes

- Formative assessment is part of formative feedback

"Information communicated to the learner that is intended to modify his or her thinking or behavior to improve learning." - Valerie Shute 2008

- Webinar focus
  - Formative assessments at scale
  - How students AND teachers can use it

Caveat: My experience as an example

# How do I use formative assessments?

Reading Quizzes and Peer Instructions

- How to make
- Policies
- How to use the data

# Reading Quizzes

- Creation
  - Question pools on assigned reading
  - 2-4 questions from previous pools (a.k.a. "Blast from the past")
  - 5-20 questions from topic(s) pool

# Reading Quizzes

- Creation
  - Question pools on assigned reading
  - 2-4 questions from previous pools (a.k.a. "Blast from the past")
  - 5-20 questions from topic(s) pool
- Policies
  - Quiz due before lecture
  - 3 chances
  - Only see whether got question correct
  - Need 75% of possible points for full credit, 3% overall grade

# Reading Quizzes

- Creation
  - Question pools on assigned reading
  - 2-4 questions from previous pools (a.k.a. "Blast from the past")
  - 5-20 questions from topic(s) pool
- Policies
  - Quiz due before lecture
  - 3 chances
  - Only see whether got question correct
  - Need 75% of possible points for full credit, 3% overall grade
- Using the data
  - Right before lecture: What questions did all the students do poorly? Do well?
  - Poorly → More time
  - Well → Less time
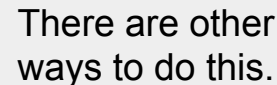
# Peer Instructions

- Creation
    - 2 identical Google forms
    - 1 - 4 questions

# Peer Instructions

- Creation
  - 2 identical Google forms
  - 1 - 4 questions
- Policies
  - Grade participation, not correctness
  - Need 75% of possible points for full credit, 3% overall grade

# Peer Instructions

- Creation
  - 2 identical Google forms
  - 1 - 4 questions
- Policies
  - Grade participation, not correctness
  - Need 75% of possible points for full credit, 3% overall grade
- During lecture
  - Students answer individually
    - > 75% get it right → explain answer
    - > 35% and < 75% → discussion
    - < 35% → hint, then discussion
  - If discussed, students answer same question again individually
  - If do poorly on round 2 → Derail lecture plan and slowly walk through what they do not understand

There are other ways to do this.

# Peer Instruction in Action! (kind of)

- Round 1: https://bit.ly/webinar-pi

# Peer Instruction in Action! (kind of)

- Round 1: https://bit.ly/webinar-pi
- Round 2: In your groups (break out rooms)
  - Optional: Introduce yourselves
  - Come to a consensus
  - Why is the right answer(s) right?
  - Why are the wrong answers wrong?
- https://bit.ly/webinar-pi-2

Online considerations

- Synchronous: Do as explained
- Asynchronous with some synchronous
  - Asynchronously: consume material and do round 1
  - Synchronously: Round 2

Still has the students:
(1) Answer by themselves,
(2) Discuss what they don't understand, and
(3) Assess what they understand after discussion

# How do I write formative assessments?

- Kinds of questions
- How to write questions

# What do I ask them?

- Simple fact check
- Predict the output of code
- What does this code do?

> What I mainly use

- Given code and an output, what was the input?
- Given code, could the function ever do X?
- Comment on code
- Compare code
- Compare code in two different languages
- Modify existing code

- Debugging
  - Find a case that reveals the bug
  - Find a case that doesn't reveal the bug
  - What class of inputs doesn't trigger the bug?
  - What class of inputs does trigger the bug?
  - Fix the bug
- Parson's problems
- Write a function that does X

Source: The CS-Ed Podcast - "Writing exams with Dan Garcia" and "Q/A with Colleen Lewis"
List available on companion blog post

# How do I write autograded questions?

Assumption: Your learning management system has a quiz feature with multiple choice and short answer question types (a.k.a. Auto-gradeable)

## Goal: Learn what the students do not understand

1. How do students usually misunderstand something?
2. Write a question(s) targeting the misunderstanding(s)
   a. Will get it wrong if have the misunderstanding
   b. If multiple-choice, each option represents a specific misunderstanding

# Example: Nested lists and pointers

```
lstA = [[0, 1], 2]
lstB = [3, 0]
lstB[1] = lstA[0]
lstA[0].append(3)
```

What is `lstB`?

A.  `[3, [0, 1, 3]]`
B.  `[3, [0, 1]]`
C.  `[3, 0, 1]`
D.  `[3, 0, 1, 3]`

Believes the inner lists are different objects, rather than pointing to the same object

Believes inner lists are unrolled when inserted into a list

Inner lists are pointing to the same object AND inner lists are unrolled

# Things to consider

- Write as many wrong answers as there are (and no more)

# Things to consider

- Write as many wrong answers as there are (and no more)
- Pay attention to the question stem
  - Highlight question's topic "Which are true?" → "Which are true about lists?"
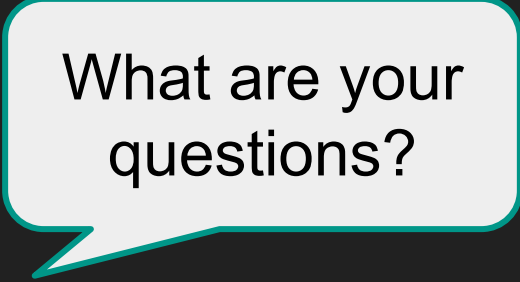
# Things to consider

- Write as many wrong answers as there are (and no more)
- Pay attention to the question stem
  - Highlight question's topic "Which are true?" → "Which are true about lists?"
- Thwart guessing

| Pick longest answer | All options similar length |
|---|---|
| Don't pick answers with "always" or "never" | No option has these words |
| If two options are opposite, one is probably the answer | Two wrong options are opposites |
| Look for keywords | Use keywords in wrong answers |
| True/False are more often True | Equal distribution or avoid entirely |

# Summary

- Defined formative assessments
- Reading quizzes and Peer instructions
  - Creation, policy, and data usage
- Process for writing questions
  - Many kinds of questions
  - Goal: Learn what the student knows

What are your questions?

Summary on blog: http://ksm-cs.blogspot.com/