PythonSneks

An Open-Source, Instructionally-Designed Introductory Curriculum with Action-Design Research

acbart.github.io/python-sneks

Slides, Field Guide, Resources

Austin Cory Bart

acbart@udel.edu

University of Delaware

Allie Sarver

Michael Friend

Larry Cox II

Virginia Tech

Plus Cal Ribbens, Panagiotis Apostolellis, John Wenskovitch, and the many folks at TLOS

Overview of Presentation

Motivating context

Experiences with Curriculum Design

Takeaways for your own courses

Slides and materials available at…

`acbart.github.io/python-sneks`

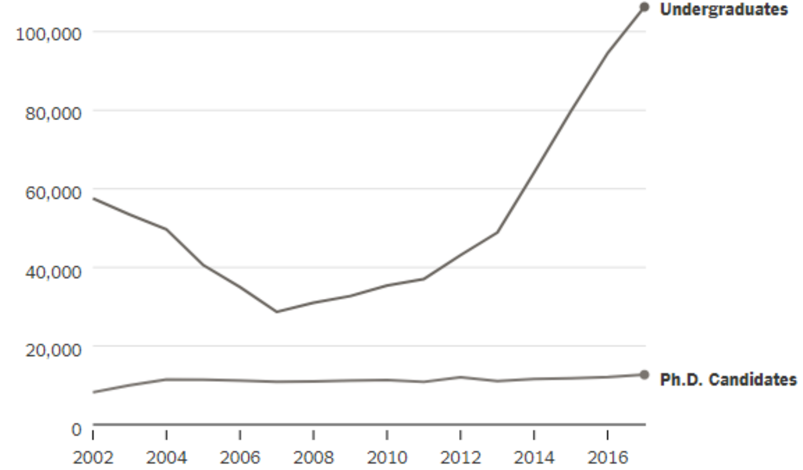
The Capacity Crisis Continues…

The New York Times

SUBSCRIBE NOW | LOG IN

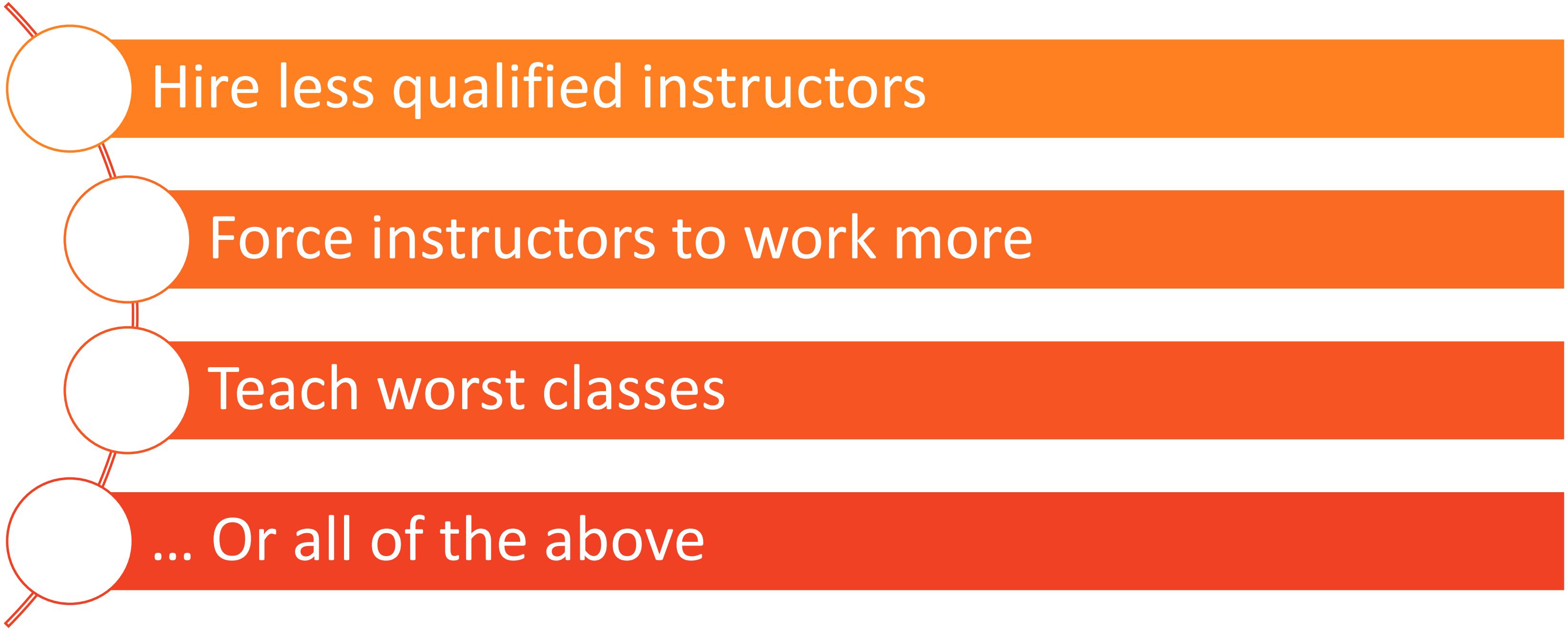
The Hard Part of Computer Science? Getting Into Class

Trend in PhDs vs. Undergrads



By The New York Times | Source: Computing Research Association Taulbee Survey

Short Term Options?



- Hire less qualified instructors
- Force instructors to work more
- Teach worst classes
- … Or all of the above

My Experience

- First year as faculty at Virginia Tech
- Need to redesign "Intro to Python for Engineers"
- Typically taught by grad students and adjuncts
- **200-300 students per section**
- 1st thought: "Wow that's a lot of students."
- 2nd thought: "I should steal existing curriculum."

What Was Available?

- Famous Curriculum
  - Media Computation
  - Earsketch
  - CT @ VT
- Open Access Courses
  - CS1001.py (Tel-Aviv University)
  - CS 61A: SICP (UC Berkeley)
  - CS 167 (Whitmann)
  - CS 1110 (UVA)
  - CS 15-112 (CMU)
  - CS1301 (Georgia Tech)

- Private Courses
  - CodeCademy
  - Various MOOCs
- Interactive Textbooks
  - How to Think Like a Computer Scientist
  - CS For All
  - FreePythonBook
- Repositories
  - CSEngageEdu
  - Ensemble

Links via "Alternative Courses" in Field Guide

But…

Usually not a full curriculum

- Some PowerPoint slides
- Sometimes programming problems

Not always suitable for my learners

- What's its difficulty level?
- What topics does it cover?

Not simple for me to reuse

- Lesson plans?
- Evaluation data?

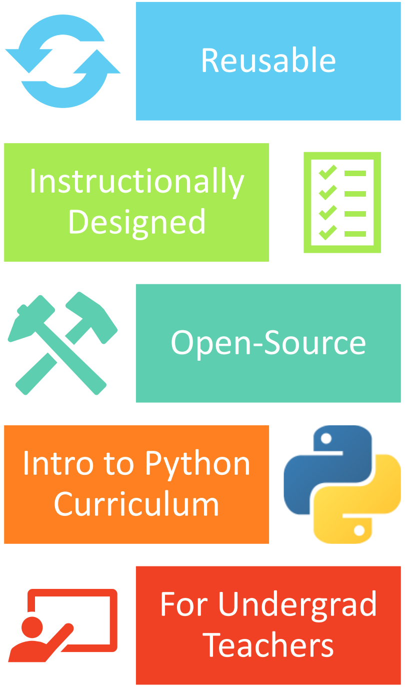
But really the *framework* I want doesn't exist!

- I want it to be easier to…
  - Create curriculum resources
  - Find curriculum resources
  - Share curriculum resources
  - Adopt curriculum resources
  - Revise curriculum resources
  - Evaluate curriculum resources
  - Pull Request curriculum resources
  - …

⚠ If you want this too, then join our Birds of a Feather #10A: The Problem of Packaging Curricular Materials

Fine, I'll do it myself!

Goal: Develop a…



Reusable

Instructionally Designed

Open-Source

Intro to Python Curriculum

For Undergrad Teachers

***PLUS** sneaky meta goal:*

Develop a *process* for designing

- Reusable
- Instructionally Designed
- Open-source
- Curricular materials
- For undergrad teachers

So What Do I Bring to You Today?

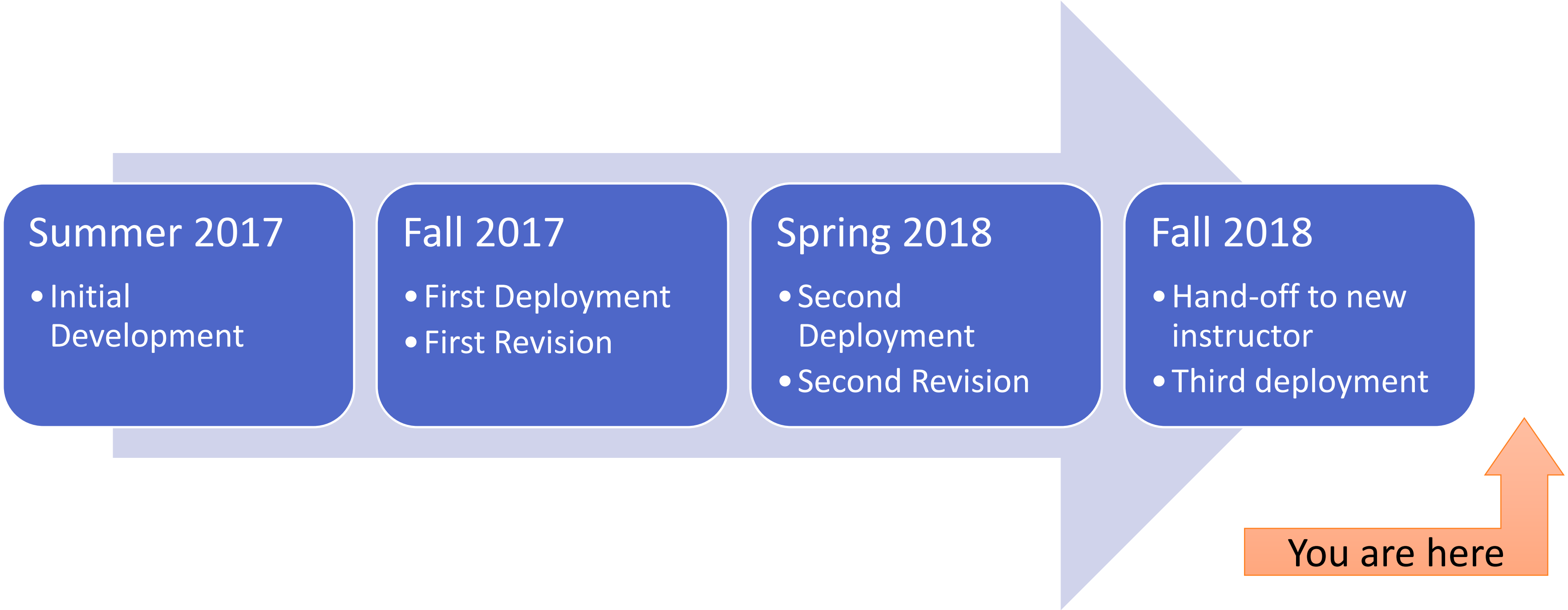
- A free Python curriculum!

`acbart.github.io/python-sneks/`

But more importantly

- How I built the curriculum
- Experiences using the curriculum
- How you can build such a curriculum

Timeline of Course



Summer 2017

- Initial Development

Fall 2017

- First Deployment
- First Revision

Spring 2018

- Second Deployment
- Second Revision

Fall 2018

- Hand-off to new instructor
- Third deployment

You are here

Summer 2017 Development

TLOS Center at VT

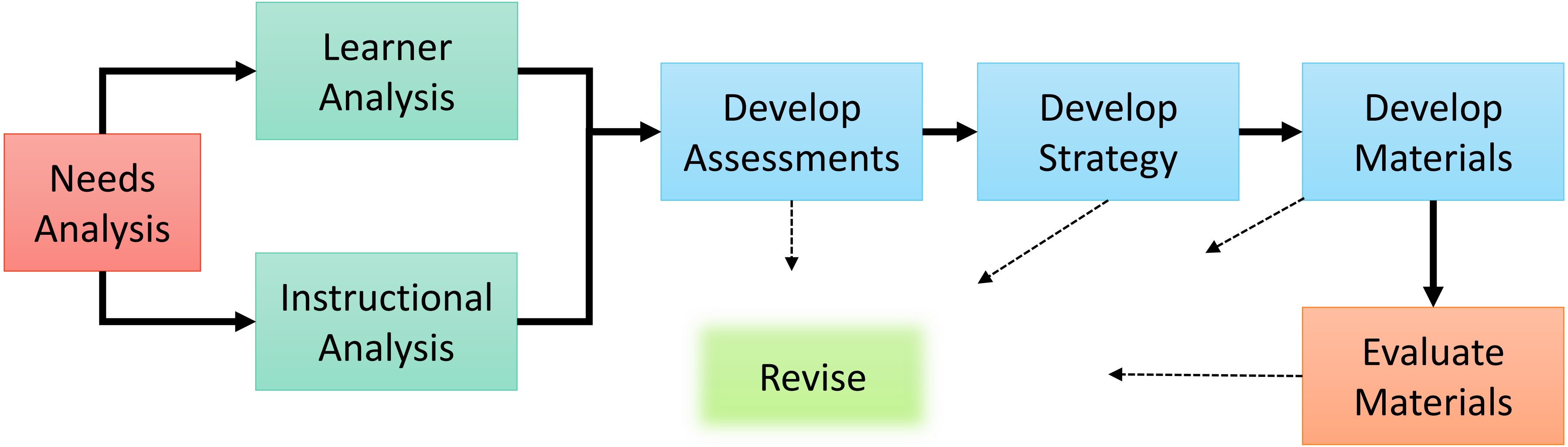
Technology-Enhanced Learning and Online Strategies

Instructional Designer

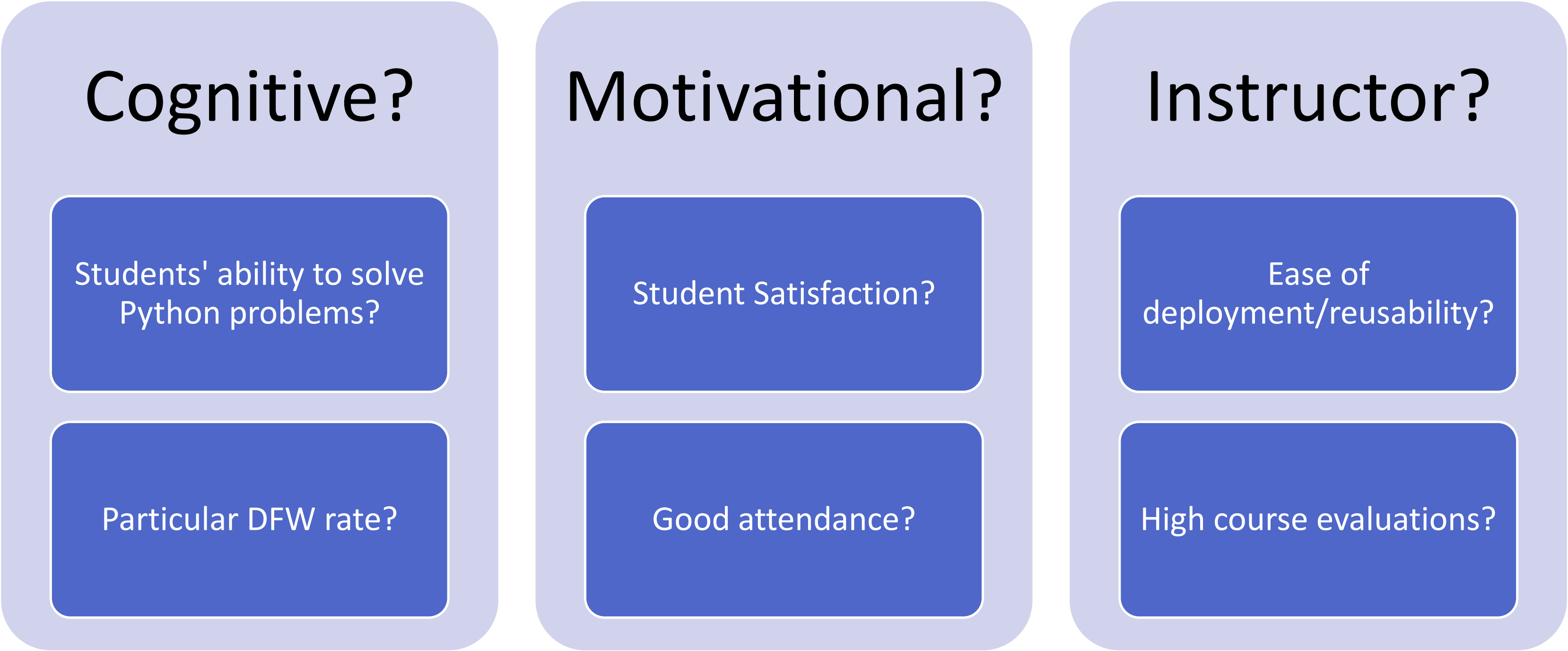
Larry Cox II

Instructional Design

A formal method of systematic course development akin to Software Engineering



Needs Analysis: Establish Goals



Cognitive?

- Students' ability to solve Python problems?
- Particular DFW rate?

Motivational?

- Student Satisfaction?
- Good attendance?

Instructor?

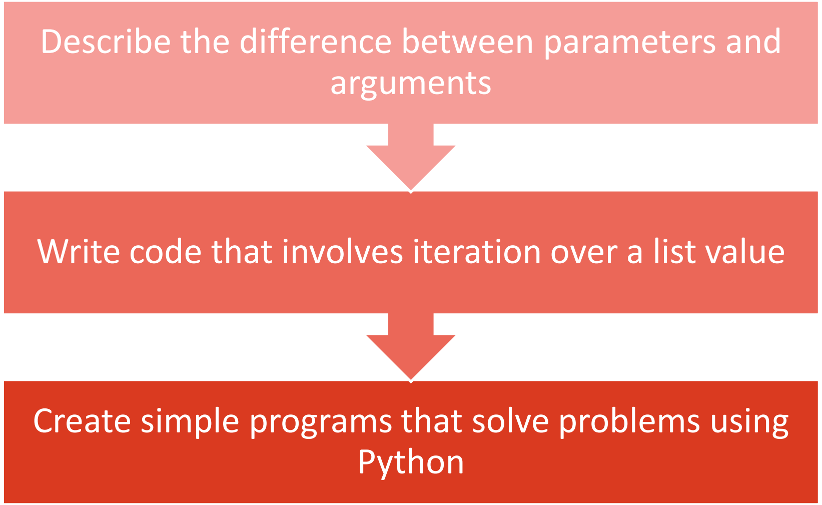
- Ease of deployment/reusability?
- High course evaluations?

Learner and Task Analyses

Who are your learners?

- 63% engineers, 5% non-STEM
- 2/3 men
- Interested, but filling a requirement
- Most have some prior experience
- Low self-efficacy

What are they doing?



Describe the difference between parameters and arguments

Write code that involves iteration over a list value

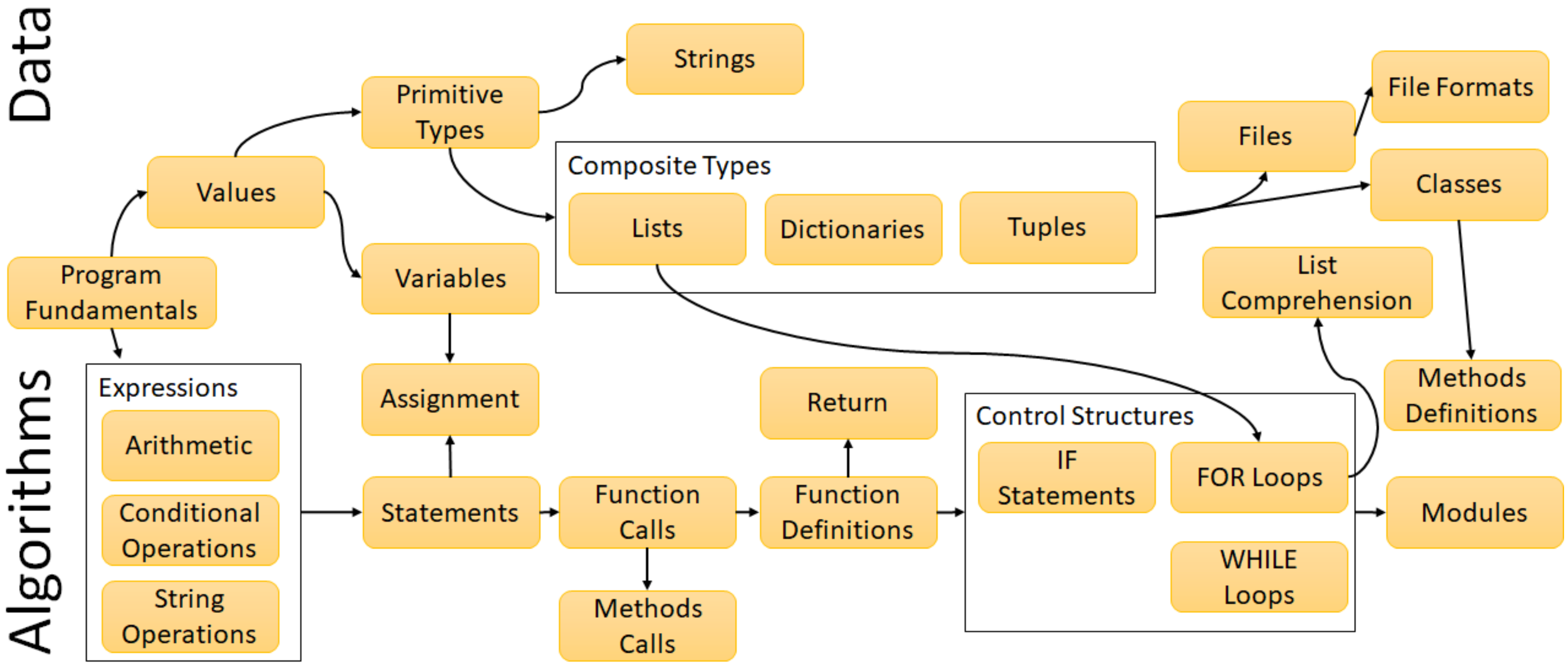
Create simple programs that solve problems using Python

234 Learning Objectives developed!

Develop Assessments

- Final exam
  - T/F, MQ, Matching: Recall terminology and concepts
  - Programming Problems: Apply skills
- Final project
  - Larger scale code organization
  - Messier, more-opened problem

Develop Strategy



Develop Materials

Mastery Quizzes

- 281 questions

Programming Exercises

- 170 problems

Video Lessons

- 51 videos

Lecture Slides

- 45 presentations

Programming Projects

- 6 projects

Ethical Reflections

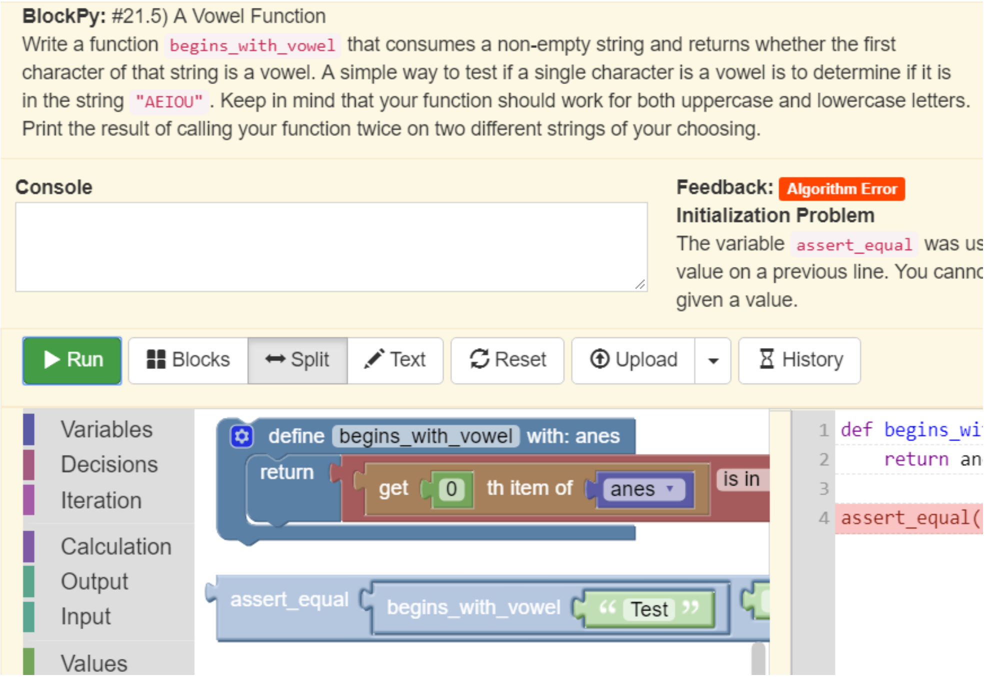
- 4 prompts

Teacher Guidebook

- 1 website

Some more detail

- 281 Mastery Quiz Questions
  - Infinite tries
  - Administered through Canvas
- 170 Programming Problems
  - Misconception-driven Feedback
  - Administered through BlockPy
- 51 Video Lessons
  - 2-4 minutes each
  - Transcript + Captions
  - Administered through YouTube



BlockPy: #21.5) A Vowel Function

Write a function `begins_with_vowel` that consumes a non-empty string and returns whether the first character of that string is a vowel. A simple way to test if a single character is a vowel is to determine if it is in the string `"AEIOU"`. Keep in mind that your function should work for both uppercase and lowercase letters. Print the result of calling your function twice on two different strings of your choosing.

Console

Feedback: Algorithm Error

Initialization Problem

The variable `assert_equal` was us value on a previous line. You cann given a value.

```
def begins_wi
    return an

assert_equal(
```

Evaluate Materials

Question Difficulty

- Classical Test Theory
- Item-Response Theory (IRT)

Essay Responses

- Rubric Scores
- SOLO Taxonomy

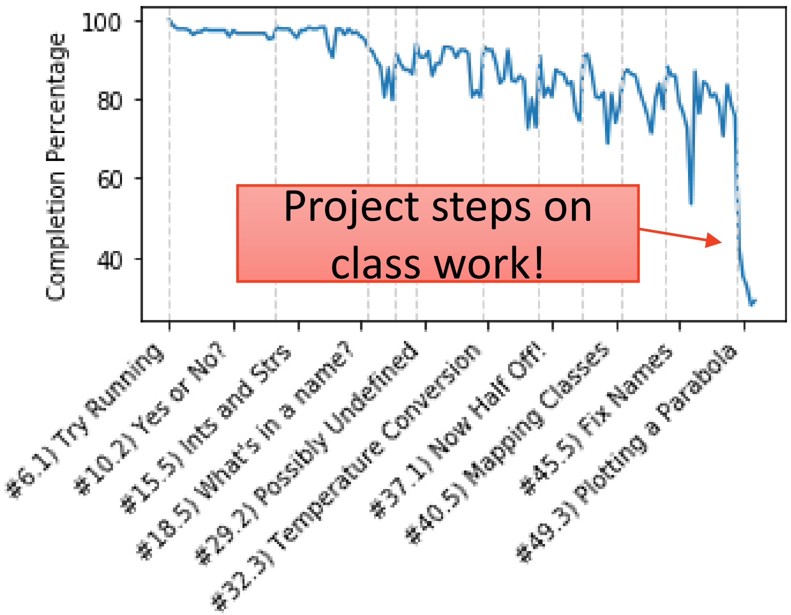
Student Self-report

- Focused survey questions
- Free-response questions

Staff Evaluations

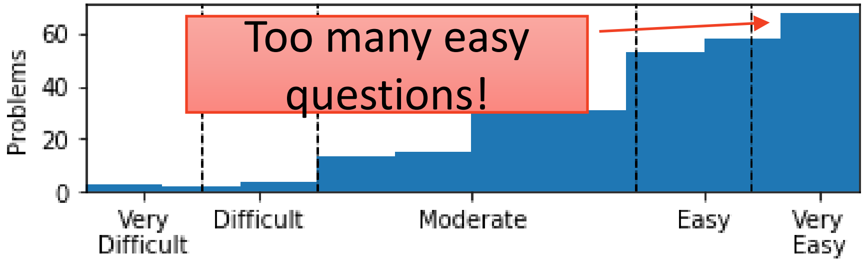
- Regular reports
- Round-table discussions

Actionable Data



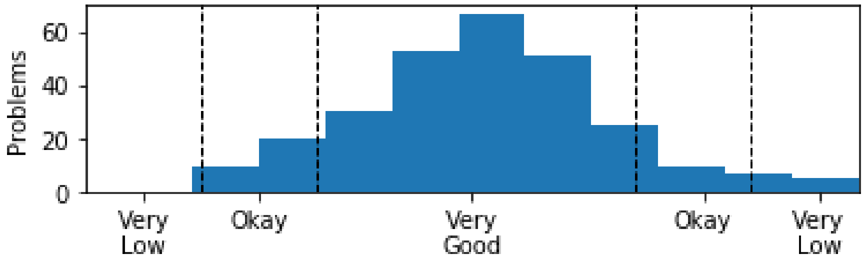
Project steps on class work!

Difficulty of Quiz Questions on First Attempt



Too many easy questions!

Discrimination of Quiz Questions for Entire Quiz



Fall 2018 Feedback *(remote deployment)*

- Quotes:
  - *"The workload on me is definitely minimal."*
  - *"I have only made small tweaks to the quizzes, and reordered perhaps 2-3 of the lessons."*
  - *"I need to update slides every week as well… but that is also a minimal time commitment, perhaps an hour on average every week."*
  - *"Overall, it's at least an order of magnitude easier than developing the course myself from scratch."*
- Of course, had issues: errors in BlockPy, pacing for diverse range of students, keeping students engaged in earlier parts of course, need more review content, etc.

Revisions

Collect concrete Issues

- Categorize by type
    - "Topical Changes"
    - "Organizational Changes"
    - "Unsolved Misconceptions"
- Prioritize
    - Low, Medium, High
- Decide what you will tackle in the time you have available

Examples:

- [Topical] [Low] Project 1 could have its strict requirements autograded.
- [Organizational] [Medium] Clearer grading style standards for projects.
- [Misconceptions] [High] Printing is not the same as returning.

Action Design Research

- The quest for unarguably valid experimental results is a deriment to teaching
- But still want to try things in a principled way
- Intervention without a real control
- Static-group comparison designs
- Acknowledges up front the limitations

Research Projects

Structured Small Groups (Allie)

- Can we promote community and better prepare students for projects by enforcing groups?

Worked Examples (Michael)

- Can we help students solve difficult problems more quickly by providing Worked Examples?

See blog posts for more detail!

Allie Sarver (sophomore) and Michael Friend (senior)

Reflections:
This was hard

- It took so much time to develop everything
  - *Hundreds* of hours over the past few years
  - Plus extra hours after leaving VT to help keep things going
- So much still to figure out
  - Public/Private/Secret data and access requirements
  - What format do others need/want?
  - How am I going to handle revisions and incorporate others' changes?

… But I think it was worth it

Reflections:
Let's Rethink Curriculum Development

- Do you follow a formal process for developing your course?
- Are your exams/materials/objectives aligned?
- What evaluations do you run?
- Can you demonstrate that the course is getting better?
- What happens when you need to teach the course a second time?
- What happens when you need to hand it off to someone else?

Reflections:
What's next?

- In Fall 2018 I moved from VT to University of Delaware
  - Teaching CS1 in Python – different audience
  - More focus on unit testing, algorithms, high-quality code
  - Want to do a Version 2
  - By the way, UD is hiring for our teaching track!
    - 5 great teaching-focused colleagues, plus me! Talk to me!
- Curriculum Materials Packaging Working Group
  - BoF #10A in Hyatt: Great Lakes A1 & A2 (4th floor)
  - Website: cssplice-cm.github.io

Conclusion

- Thanks to many folks for making this possible
  - Cal Ribbens
  - Larry Cox II
  - Allie Sarver
  - Michael Friend
  - Panagiotis Apostolellis
  - John Wenskovitch
  - Virginia Tech TLOS
- And you!

Curriculum Materials Packaging
Birds of a Feather #10A
Hyatt: Great Lakes A1 & A2
`cssplice-cm.github.io`

acbart.github.io/python-sneks