LC Course Staff Training

Curriculum & Best Practices

Agenda

1. How LC Creates and Delivers Content
2. Coursework Requirements
3. Understanding and Supporting Learners
4. Putting It Into Action

How LC Creates and Delivers Content

Blended Learning In a Nutshell

- Activities are structured and assigned in order to **maximize student outcomes**
- Tasks that can be completed independently are done **outside of classroom learning time**
  - Reading, watching video tutorials, basic practice
- Tasks that are more difficult or require support are done **in class**
  - Students benefit from real-time feedback from instructors, TAs, and fellow students
  - More support reduces the stress of a difficult activity and encourages risk-taking

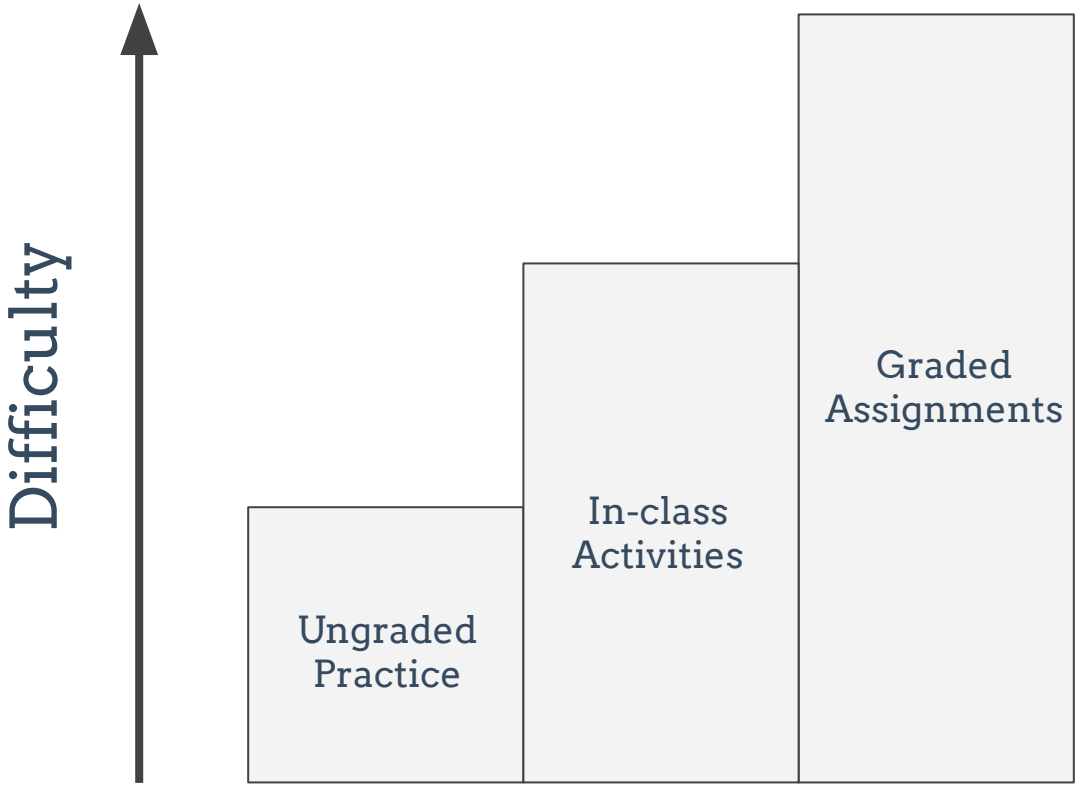
Why Does LC Use Blended Learning?

- Prep work then practice.
- Optimize the time we spend with the students.
  - Start building relationships as soon as possible
- It encourages students to be active, independent learners while also giving them the support that they need.
- Enables students with different needs and preferences to have control of their own learning experience.
- It works!

Remote Blended Learning

- Blended learning hinges on in-person interactions in a classroom. This means that our remote class time is especially important to students' ability to learn.
- In a remote learning environment, we will need to encourage students to be active, independent learners while also giving them the support they need.
- We must enable students with different needs and preferences to have control of their own learning experience online.
- Students absorb the content presented at the beginning of class time the most.

Learning By Doing



Commonly Used Class Platforms

- GitHub and GitLab
  - Eventually, students may use Git and GitHub to turn in assignments.
  - Solutions can be made available to TAs on GitLab.

- Slack
  - All students, course staff, and some LaunchCode staff will communicate in a class specific Slack group.

- Canvas
  - The source of truth for the course.

Coursework Requirements:

Demos, Grading, & Solutions

Assignment Demos - they have to happen

- Assignments - and attendance - are the **only** items required to pass
- Writing code and explaining code are two different skills
  - We want our graduates to leave our courses with both
- Job-readiness includes talking about code you've written and assignment walkthroughs give practical experience
- Brief, one-on-one interactions with students give you more insight into who is struggling and how to help them

Assignment Demos - how to run them

- After a student has submitted an assignment, you must schedule time for a demo before marking their grade in Canvas
- Review the provided rubrics in GitLab so you know what the requirements are
- Start by asking the student to demonstrate the working program, after you verify the output works as expected, it's time to 'look under the hood' and have them go over the coding work they did to achieve those results

Questions

Grading:

GitHub Classroom Walkthrough

GitHub Classroom

- Creates a repo for the student's assignment that holds and runs the auto-grading tests.
  - Course Staff see all repos, can add comments in the form of a PR, can compare commits, see which tests failed, etc.
- Students never see beyond their own repo.
- Quick glance see progress of students on assignments
  - Green Check means Passed
  - Red Check has not passed yet(failed)

Assignment #3: TechJobs (MVC Edition)

Individual assignment · Active

https://classroom.gi1 | Edit | Download

Accepted students	72
Assignment submissions	0
Passing students	45/72

Search by GitHub username or student identifier

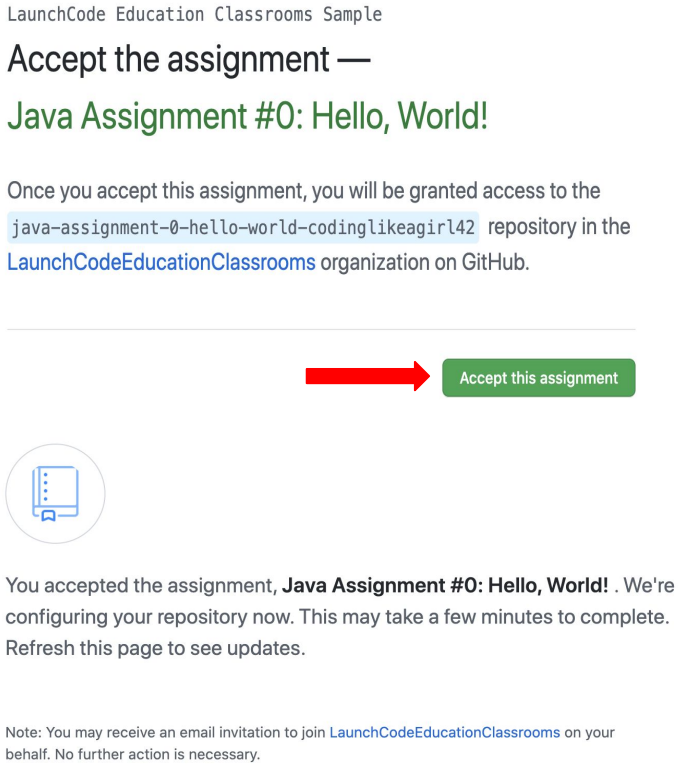
Total students — Submitted · Passing · Sort

Student 1	Latest commit failed	0/1	5 commits
Student 2	Latest commit passed	1/1	8 commits

What a student sees:

- When students click on link in canvas and accept classroom assignment

- GitHub automatically creates a repo for them.



Solutions:

A GitLab Walkthrough

GitLab - Exercises, Studios, and Assignments

- Contains solutions for all exercises, studios and assignments by learning track.
  - Rubrics are included for all graded assignments.

- Each learning track is organized differently, but contains all necessary solutions
  - *FYI:* LC101 is the JavaScript unit

- **Note:** Remember there might be more than one way to code something

Questions

Break!

Understanding and Supporting Learners

Mentoring

Who students turn to when they need help:

- Code questions, installation issues, device troubleshooting:
  - Classmates then TAs (and finally instructor for LaunchCode cohort)
- Due dates, Canvas issues, class modifications/interventions:
  - TAs and Candidate Engagement Manager
- Curriculum edits and starter code issues:
  - TAs and LaunchCode (as applicable)
- Other:
  - ***TAs***

Student support

Not all questions should be answered the same way.

- As much as possible, model how we want students to approach solving a problem.
- Encourage independence, research, and reflecting on attempts to solve a problem. The Socratic Method works.
  - Let them struggle for a little while
- However, be ready to step in if a student gets deep into the weeds or if their frustration is preventing progress.
- Empathy, experience, a short break, snacks/water...

Technical vs. Non-technical Discussions

- BE HONEST
  - Talk about your knowledge gaps
  - Talk about your experiences
- Acknowledge the student's feelings
  - Frustration when it comes to technical work is valid
  - Feeling unsure about the future is valid
- Reinforce that the perception of others' knowledge is just a perception

Sample Online Interaction

One student dominates the conversation during in-class activities.

Sample Online Interaction

Student does not interact in any way, has camera off

Sample Online Interaction

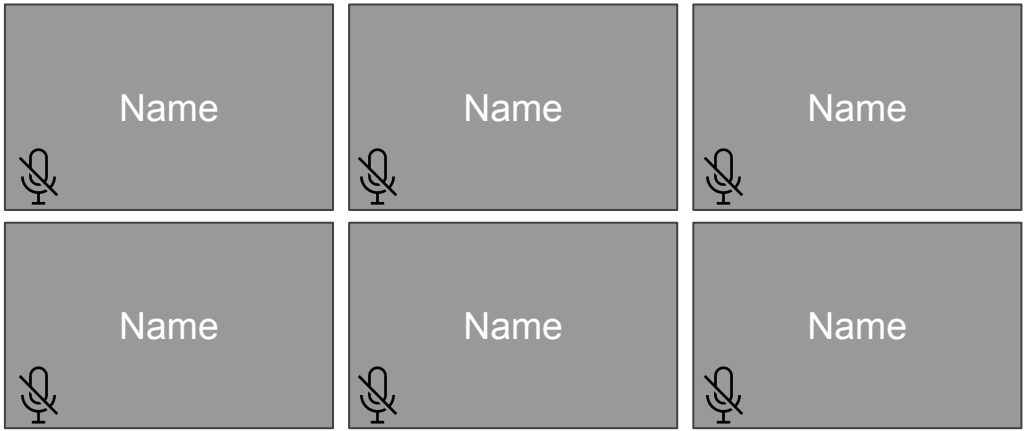
Student is silent for long stretches of time, but is still present and speaks up when they need something. (Do they need help?)

Sample Online Interaction

Student needs validation for every step.

Sample Online Interaction

This Zoom screen for the whole class:



Student support
Wrap up

You don't have to wait for a question!

What do they know vs. what do they need to know?

How to Google an error

Slack as a peer support network

Group work/pair programming

Temperature Check
&
Break

Putting It Into Action

How To Mentor Online

- Online teaching can be difficult. The first class matters!
- Start strong with clear expectations for yourself and for students
- Continue engagement and boundary-setting throughout the course
- Be willing to share!
- Make a *prepare to teach* checklist
- Good distractions (Home-life can seep into online class).
- Activity time checklist
- Ideas for sparking discussion?

Getting ready for class

- Prep work is not just for the students!
- Review (or complete) activities yourself *before* class starts.
- Anticipate where your students might make common mistakes.
- Check Slack to see if your students asked for feedback or assistance on anything.

In-class Best Practices

1. Check in with EVERY student during EVERY activity.
    a. Builds relationships and trust.
    b. Checking in does NOT mean a quick question like, "How are you doing? Need any help?" Ask specific questions about their code, their interpretation of the instructions, or their coding dreams.
    c. Provide detailed feedback as often as possible. "Good job!" does not cut it.
2. Be ready to clarify instructions beyond just re-reading the words on the screen.
    a. Try a different analogy
    b. Break it down into smaller parts
3. Encourage students to work together and share ideas.
4. Assist individuals as questions arise. Address frequent mistakes and/or questions to your whole group.
5. Make a note of any issues that occur during the activity and provide that feedback to the rest of your TAs and/or LaunchCode team.

Questions