Project Courses

Prasun Dewan (dewan@cs.unc.edu)
Department of Computer Science
University of North Carolina at Chapel Hill

What structure do you expect?

So I can adapt

So you think abstractly about the structure of a typical lecture

So I can learn from you

Lecture Goals

- Definition of Project
- Impact on Teaching Activities
- Techniques
- Tools
- Principles/Data

Projects vs. Non-Projects

- Compute the product of two numbers ✗
- Write an essay ✓
- Describe body's response to a health problem ✗
- Classify a species ✗
- Assemble a device ✓
- Create a software program ✓
- Theorem proof ✓

- Tangible "Virtual or Physical" Artifact
- Real-World Problem
- More than one Solution to Problem
- Industrial-Strength Tools
- Degree of Projectess!

Krajcik, J. and N. Shin (2014). Project-Based Learning. The Cambridge Handbook of the Learning Sciences. R. K. Saw
Cambridge University Press: 275-297.

Pedagogy Styles

Compute the product of two numbers

Write an essay

Describe body's response to a health problem

Classify a species

Assemble a device

Create a software program

Theorem proof

Inductive

Deductive

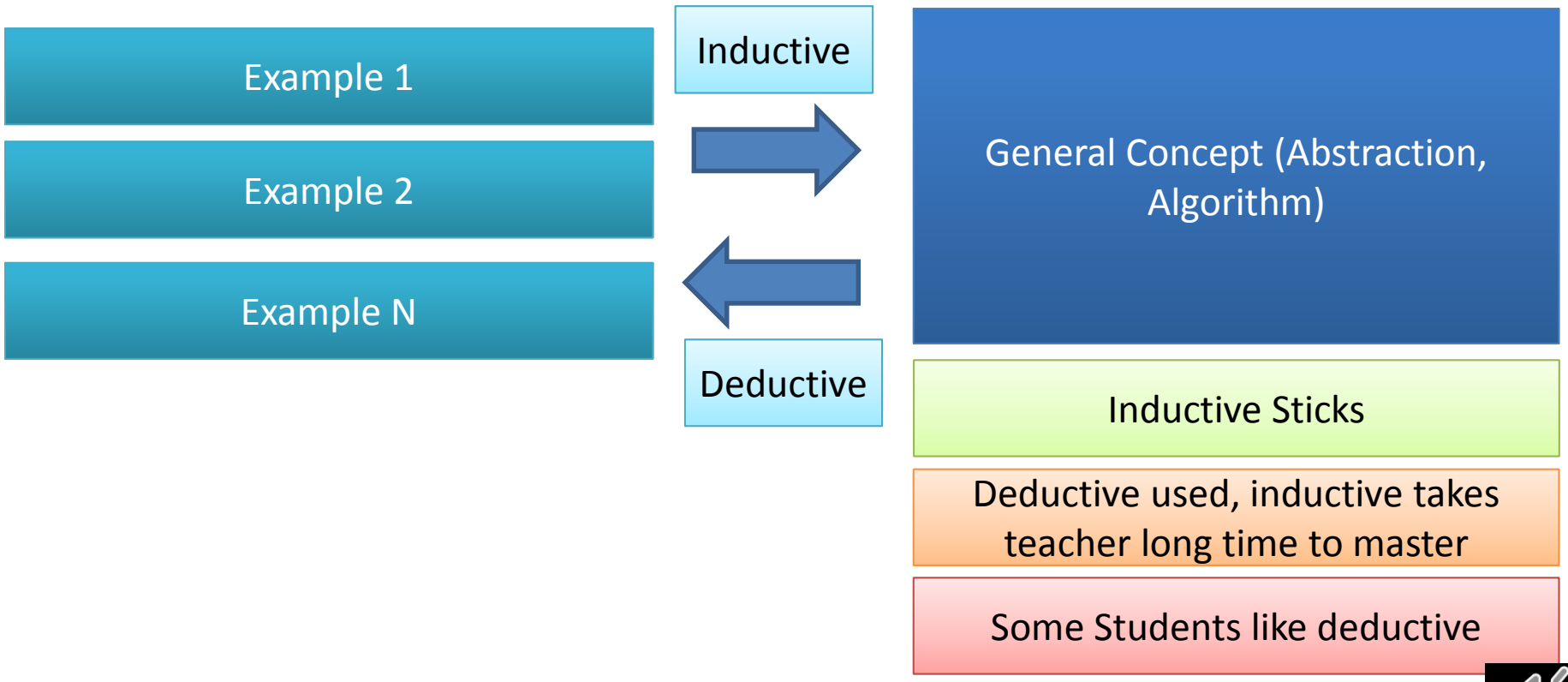
Tangible "Virtual or Physical" Artifact

Real-World Problem

More than one Solution to Problem

Industrial-Strength Tools

Inductive vs. Deductive



Felder, Richard M , **Reaching the Second Tier--Learning and Teaching Styles in College Science Education.**. *Journal of College Science Teaching*, v22 n5 p286-90 Mar-Apr 1993

Maximum “Projectness” in CS

Intro to Programming

Theory of Computation

Algorithms

Compilers

Software Engineering

Compilers and Software Engineering can be taught without projects

Project-based learning is a principle easier applied to certain courses

Cumulative assignments addressing real-world problems using real-world tools on creating a semester-long project

Why cumulative?

Distributing the Work and Feedback

Assignment 4

Assignment 3

Assignment 2

Assignment 1

"Distributed" learning is more effective than "massed" learning

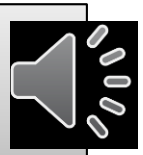
Incremental tutor feedback while doing assignment helps learning

Peer feedback in active learning helps learning

Kirkley, J. (2012). Distributed Learning. Encyclopedia of the Sciences of Learning. N. M. Seel, Springer: 1020-1021.

Roger Azeveo, Daniel C. Moos, Jeffrey A. Greene, Fielding I. Winters, Jennifer G. Cromley(2008). "Why is externally-facilitated regulated learning more effective than self-regulated learning with hypermedia?" Educational Technology Research and Development **56**(1): 45-72..

Sarah L. Eddy* and Kelly A. Hogan (2014). "Getting Under the Hood: How and for Whom Does Increasing Course Structure Work?" CBE-Life Science Education



Principles

Student Efficiency

Instructor Efficiency

Teaching is Design!

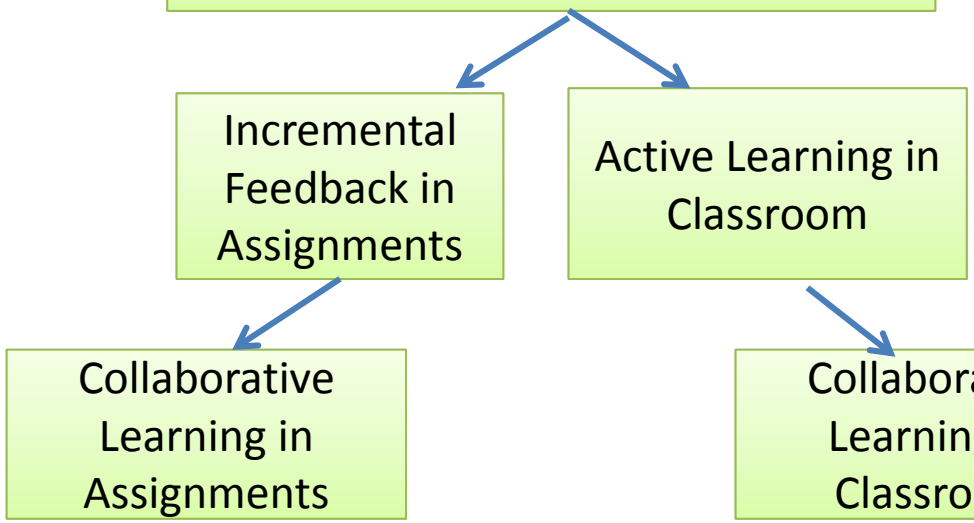
Programming Projects make it easier or harder to make these tradeoffs?

What is Special about Programming?

Project-based Learning

Inductive Learning

Distributed Learning



Programming vs Other Projects

Composing a Lecture	Programs can be manipulated by other programs more easily
Create a Software Program	Easier to tell if they are not right

Lecture Goals

- Definition ✓
- Impact on Teaching Activities ✓
- Techniques
- Tools
- Principles/Data ✓

Discipline-Independent Approach: Quiz + Post-Quiz

Question

The size of an array (object) can change dynamically during program execution (Please do not confuse an array object with an array variable, which can be assigned different array (objects).

- ○ True
- ○ False

Reason for Confusion

```
int[] assignmentScores = {100, 98, 99, 100, 90, 80};

assignmentScores = new int[] {60, 40, 50};
```

Post-quiz collaboration can resolve confusion

On its own, not project-based learning

Sarah L. Eddy* and Kelly A. Hogan (2014). "Getting Under the Hood: How and for Whom Does Increasing Course Structure Work?" CBE-Life Science Education

CS: Discovery-Promoting Praxis

```
System.out.println ("Length of uninitializedElements: "
        + uninitializedElements.length);

// Put break point below (for later in the praxis)
uninitializedElements[0] = new ARecursiveFactorialSpreadsheet();
uninitializedElements[0] = new ALoopingFactorialSpreadsheet();
System.out.println (uninitializedElements.length);
System.out.println("Contents of unininitializedElements:" +
        Arrays.toString(uninitializedElements));

// Let's set uninitializedElements to point to a different, new array
uninitializedElements = new FactorialSpreadsheet[] {
        new ALoopingFactorialSpreadsheet()
};
System.out.println ("Length of uninitializedElements now: " +
        uninitializedElements.length);
System.out.println("Contents of unininitializedElements:" +
        Arrays.toString(uninitializedElements));
/*
 * (T/F) The size of an array can change dynamically during program
 * execution.
 *
 * This is tricky. Did either of the arrays assigned to uninitializedElements ever change
 * size in this program? Keep in mind that these are two different arrays we are talking about.
*/
```

Debugger and program output provide some answers learned inductively

They play the role of a collaborator in previous case

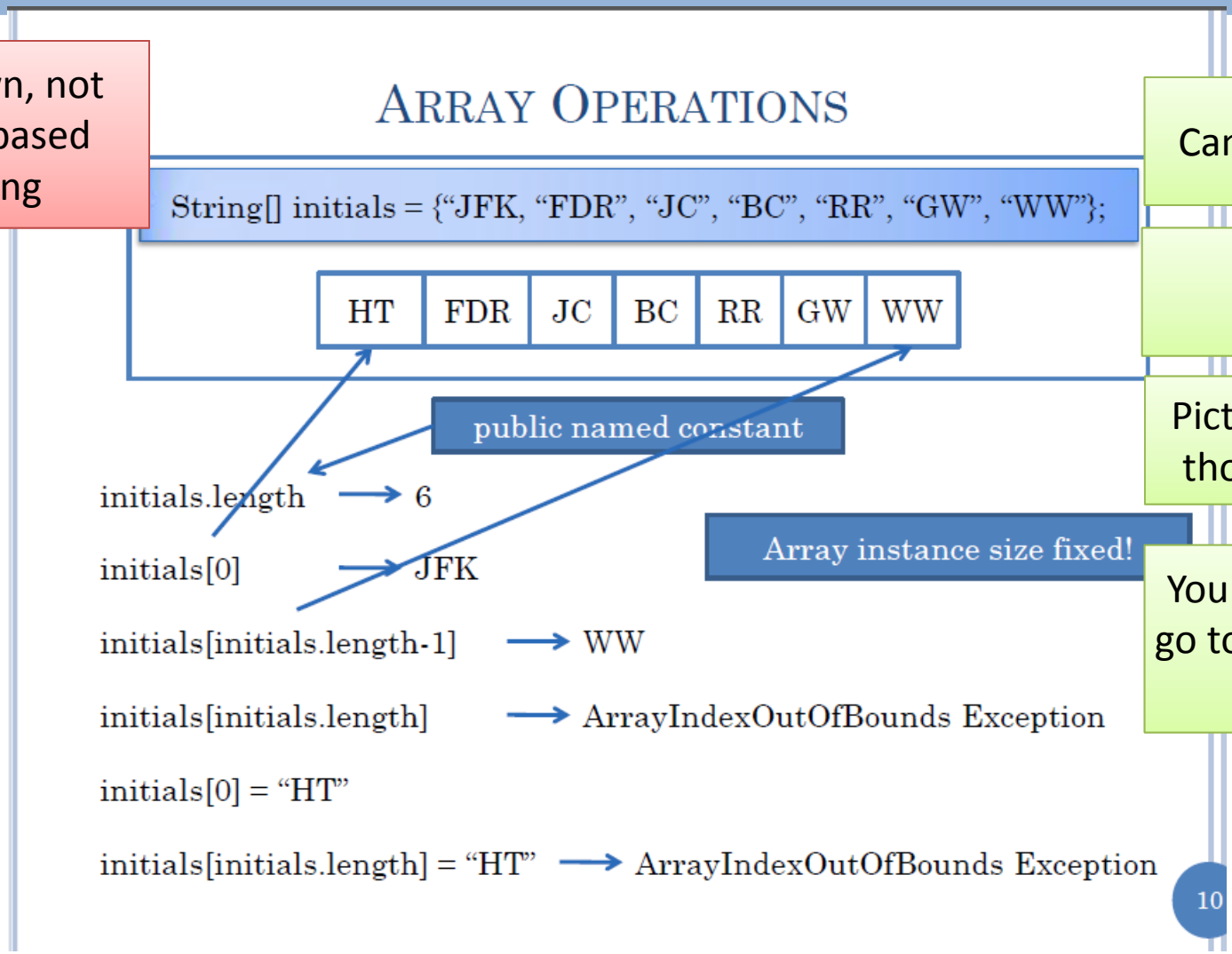
“Teach a person to fish”

Project-based learning

Alternative to lecture, not a step after it

Live/Recorded Lecture

On its own, not project-based learning



Can be inductive

Efficient

Picture is worth a thousand words

You do not have to go to war to know it is wrong

Hybrid Approach with Quiz-Assignment Transaction

Quiz

Assignment

Deliverable

Recorded Lecture

Praxis

Leaning Means

Makes sense to view the recorded lecture after doing the assignment?

Makes sense to do the quiz after the assignment?

Class time is used to view lecture, do praxis and sometimes ask assignment questions

Lecture Goals

- Definition ✓
- Impact on Teaching Activities ✓
- Techniques ✓
- Tools
- Principles/Data ✓

Discovery-based Praxis

```
System.out.println ("Length of uninitializedElements: "
        + uninitializedElements.length);

// Put break point below (for later in the praxis)
uninitializedElements[0] = new ARecursiveFactorialSpreadsheet();
uninitializedElements[0] = new ALoopingFactorialSpreadsheet();
System.out.println (uninitializedElements.length);
System.out.println("Contents of unininitializedElements:" +
        Arrays.toString(uninitializedElements));

// Let's set uninitializedElements to point to a different, new array
uninitializedElements = new FactorialSpreadsheet[] {
        new ALoopingFactorialSpreadsheet()
};
System.out.println ("Length of uninitializedElements now: " +
        uninitializedElements.length);
System.out.println("Contents of unininitializedElements:" +
        Arrays.toString(uninitializedElements));
/*
 * (T/F) The size of an array can change dynamically during program
 * execution.
 *
 * This is tricky. Did either of the arrays assigned to uninitializedElements ever change
 * size in this program? Keep in mind that these are two different arrays we are talking about.
*/
```

Not solving a problem, degree of projectness is low

Grading Programming Problems

Problem: Write a recursive factorial function

Tools to grade it?

How do we adapt the problem to use the tool?

Consequences of using the tool?

Limits to Automatic Grading

Problem: Write a recursive factorial function

Source-code analysis runs into halting problem

Runtime testing checks a subset of the I/O pairs and cannot check for style such as recursion

Diffing Output Given an Input

Number?
3
Factorial: 6

Number?
3
Factorial: 3

Please enter a number?
3
Factorial: 6

Used in Udacity for small assignments but not projects

Not tolerant to Inconsequential, intuitive mistakes in I/O

Allow tool to be invoked before submission

Student can hack solution and hardwire solution to test cases

Students do not have access to all tests

Reduced creativity

Reduces natural divergence and hen plagiarism detection

Unit Testing

```
public void testFactorial () {
   Assert.assertTrue("3! = 6", Factorial.fact(3) == 6);
}
```

```
package functions;
public class Fact {
   public static factorial (int n) { …. }}
```

Not tolerant to Inconsequential, intuitive mistakes in naming, parameter order and organization

Allow tool to be invoked before submission

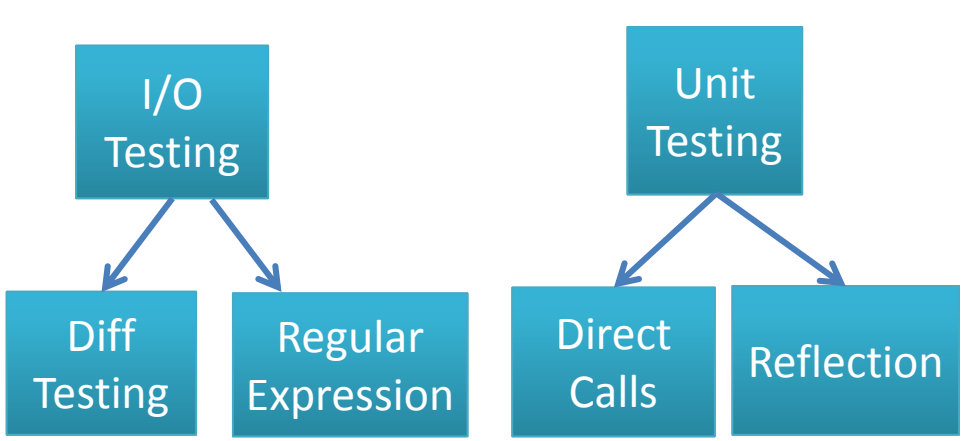
Student can hack solution and hardwire solution to test cases

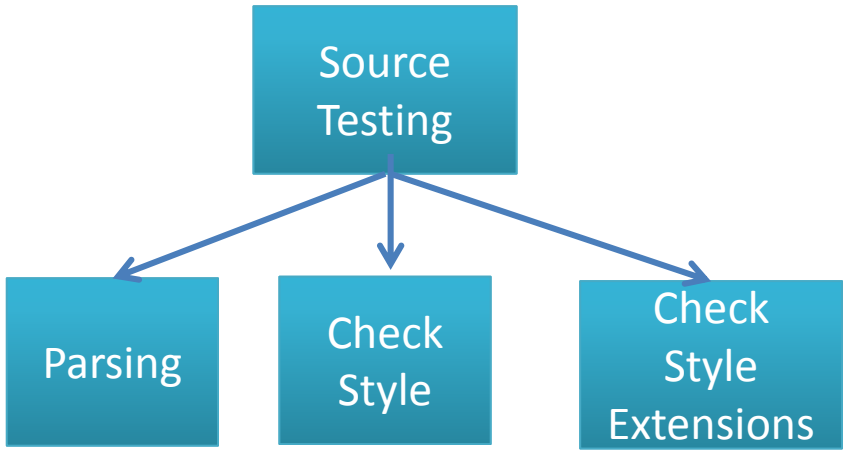
Students do not have access to all tests

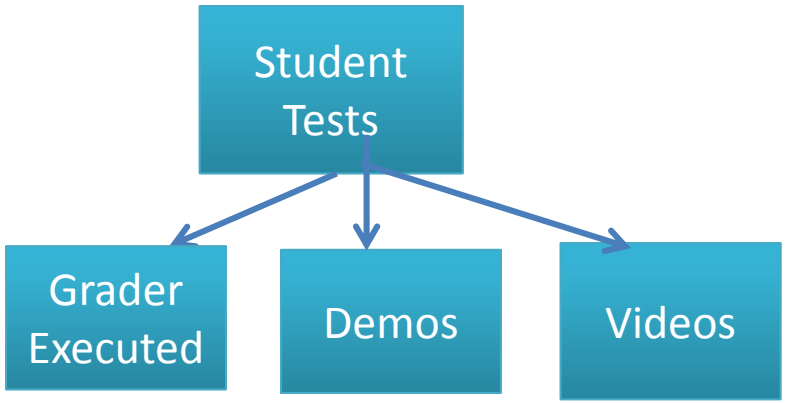
Reduced creativity

Reduces natural divergence and hen plagiarism detection

Multi-Faceted Approach





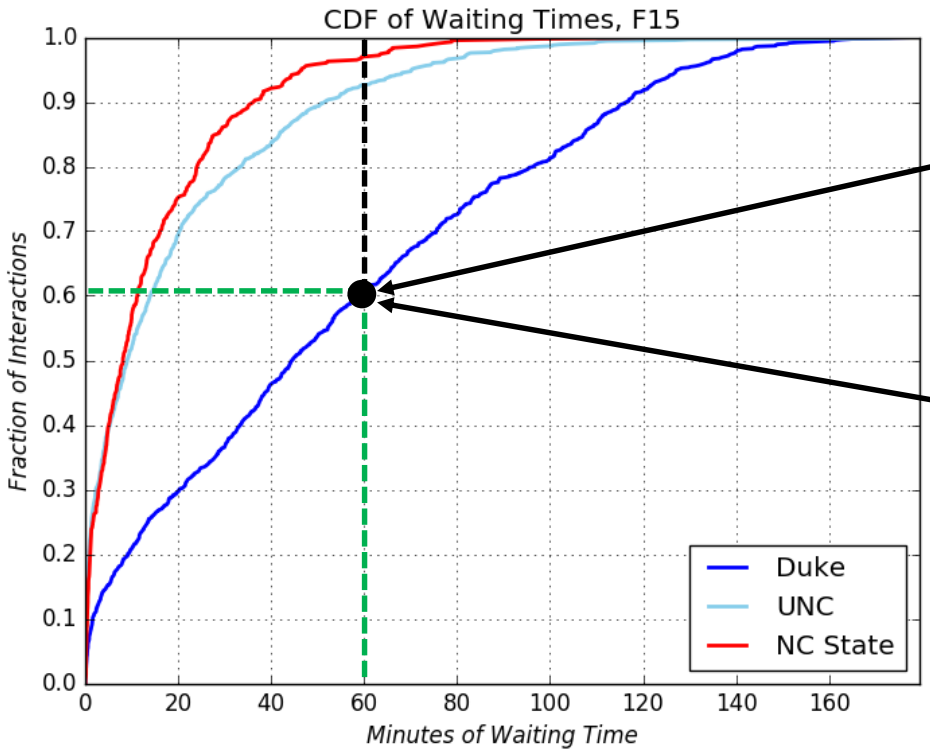


Rubric Management

Learning Management Systems

Halting problem opens a whole new research area

Excessive Waiting Times



40% of the time, students wait for over an hour

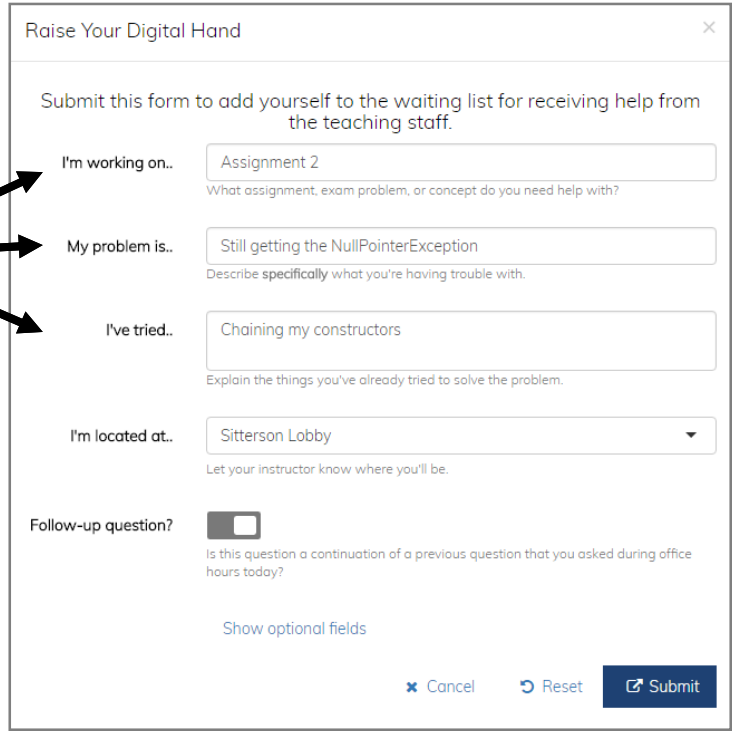
60% of the time, students wait for under an hour

Slide by Aaaron Smith

Feature: Student Question Prompts

The “raise hand” form prompts students to think critically about their issue before asking for help

Student “Raise Hand” Form



Slide by Aaaron Smith

Collaboration on Design

Collaboration Allowed and Honor Court

1. You are encouraged and expected to discuss the assignments among yourselves.
2. You are permitted to discuss all aspects of the Java programming language with anyone.
3. You are permitted to discuss solutions at the design level but not at the code level. For example, you are allowed to tell others that you have separate

---

classes in your program for scanning and evaluating expressions, or that you are using a loop instead of recursion for scanning, but are not allowed to show them Java interfaces, classes, while loops or other Java code in your solution. A general rule of thumb is that if you are communicating using a natural language, you are discussing at the design level, but if you are communicating using pseudo or actual code, you are discussing at the code level.

Give Design Away (in small print)

Part 1: Rotating Fixed Line

Create a class that implements a line shape that can be rotated around the Java origin (0, 0). The upper left corner of (the bounding box of the line) is always the Java origin. The lower-right corner of the line is always a fixed distance from the origin and can be rotated based on its current angle.

The line should be displayable by ObjectEditor. This means it must have the line properties and annotations expected by ObjectEditor. As the upper left corner is fixed, the line class does not have setters for the location of this point. It also need not have setters for the height and width properties of a line. It should have public methods for setting the radius and angle of (the lower-right corner) of the line. These methods take double values determining the absolute radius and angle.

In addition, the class must have a method to change the angle of the line by a certain amount. This method must take an int argument. You are free to determine the appropriate scale. For example, you might decide that one int unit corresponds to Math.PI/32. In this case, rotating the line by 16 units adds 90 degrees (Math.PI/2) to its angle. This method must call the angle setter.

Try to implement this class on your own before you read the remainder of this paragraph. I believe the easiest way to implement such a line is to declare an internal instance variable that stores the current lower-right corner in an instance of the class APolarPoint we saw in lectures. This variable is not exported as a property to ensure ObjectEditor does not display it. The getter for the height and width property of the line depend on the value assigned to this variable, and the setters for the radius and angle of the line assign a new immutable instance of APolarPoint to the internal variable.

Students Getting Behind

Assignment 4	Assignment 4 Due Date
Assignment 3	Assignment 3 Due Date
Assignment 2	Assignment 2 Due Date
Assignment 1	Assignment 1 Due Date

What if you get permanently behind?

Can shift assignment dates N times if last N assignments will not be done.

But you sacrifice the last N assignments, whose scores will go in fudge factor.

Summary

- Principles
  - Project-based learning, Inductive learning, Distributed learning, Active-learning, Incremental Feedback, Collaborative learning, Student and Instructor Efficiency
- Techniques
  - Discovery-enabling Praxes as an Alternative
  - Quiz/Assignment Transaction
  - Give design away
- Tools
  - Graders based on diff and unit tests
  - Reduce project-based learning and conflict with plagiarism (detectors)
  - Can be mixed and integrated with manual and semi-automatic techniques
  - Research area
- Design space and tradeoffs
  - Constantly evolving and challenging design activity