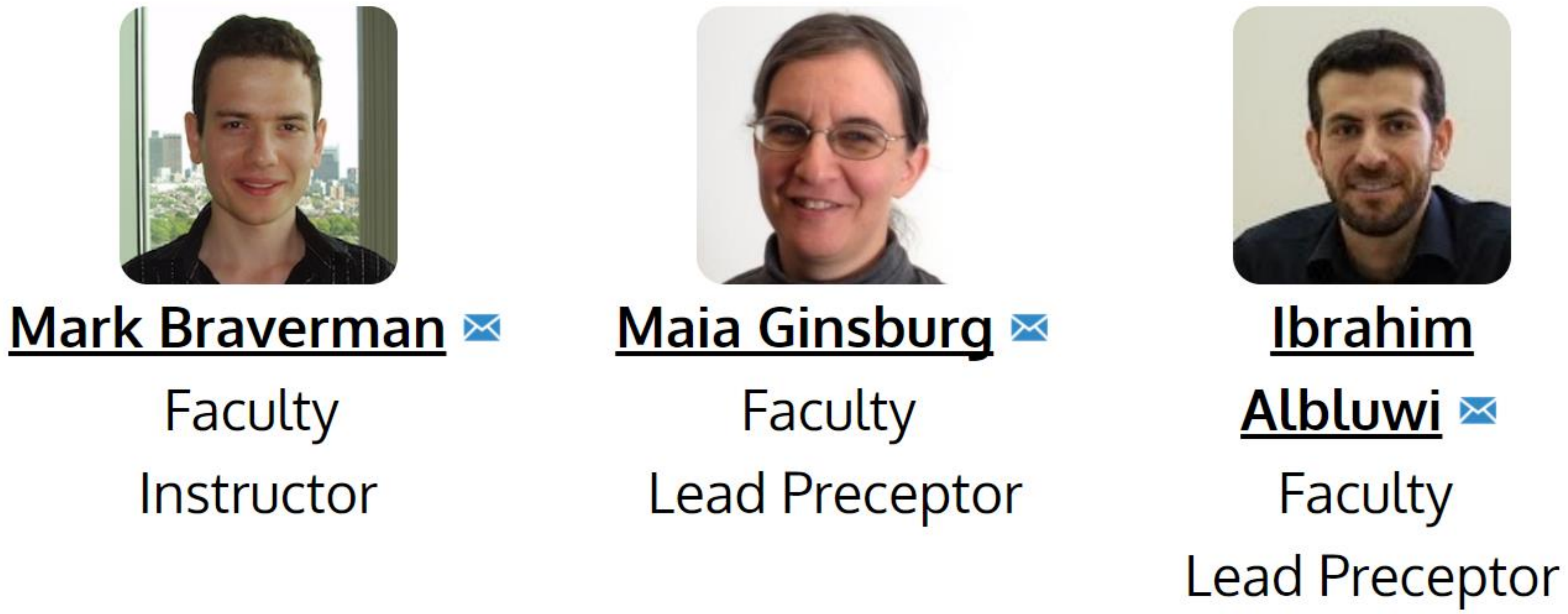
ALGORITHMS

by Robert Sedgewick and Kevin Wayne

PRINCETON — COS 226
Spring 2018
Taught by Ibrahim Albluwi, Mark Braverman, and Maia Ginsburg



Mark Braverman ✉
Faculty
Instructor

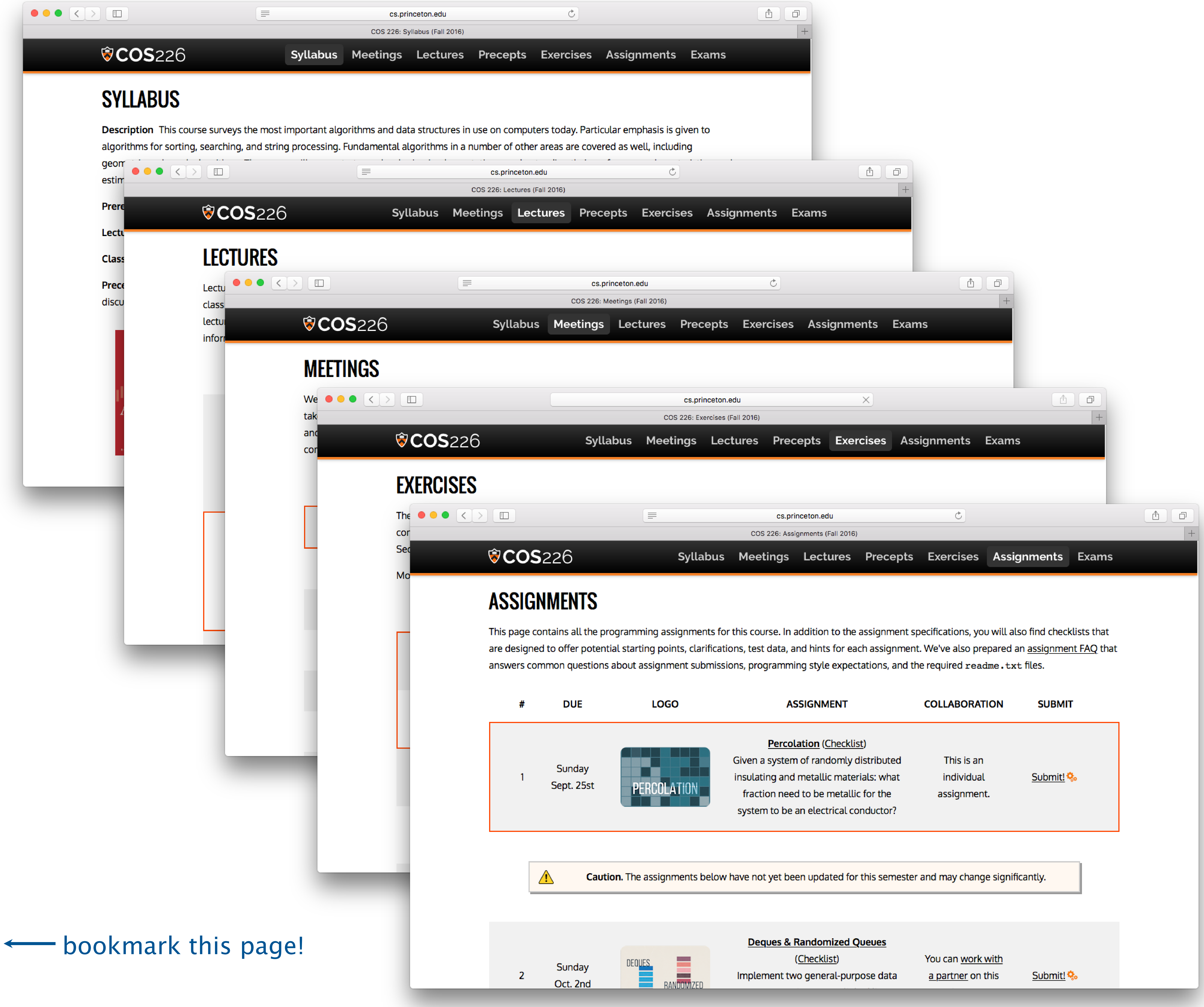
Maia Ginsburg ✉
Faculty
Lead Preceptor

Ibrahim Albluwi ✉
Faculty
Lead Preceptor

Course website

COS 226 website.

- Syllabus.
- Lectures.
- Meetings.
- Exercises.
- Assignements.
- Precepts.



cs.princeton.edu
COS 226: Syllabus (Fall 2016)
COS226
Syllabus
Meetings
Lectures
Precepts
Exercises
Assignments
Exams
COS 226: Lectures (Fall 2016)
COS 226: Meetings (Fall 2016)
COS 226: Exercises (Fall 2016)
COS 226: Assignments (Fall 2016)

SYLLABUS

Description This course surveys the most important algorithms and data structures in use on computers today. Particular emphasis is given to algorithms for sorting, searching, and string processing. Fundamental algorithms in a number of other areas are covered as well, including geom estim

Prere

Lectu

Class

Prece discu

LECTURES

Lectu class lectu infor

MEETINGS

We tak and cor

EXERCISES

The cor Sec

Mo

ASSIGNMENTS

This page contains all the programming assignments for this course. In addition to the assignment specifications, you will also find checklists that are designed to offer potential starting points, clarifications, test data, and hints for each assignment. We've also prepared an assignment FAQ that answers common questions about assignment submissions, programming style expectations, and the required `readme.txt` files.

#	DUE	LOGO	ASSIGNMENT	COLLABORATION	SUBMIT
1	Sunday Sept. 25th	PERCOLATION	**Percolation** (Checklist) Given a system of randomly distributed insulating and metallic materials: what fraction need to be metallic for the system to be an electrical conductor?	This is an individual assignment.	Submit!

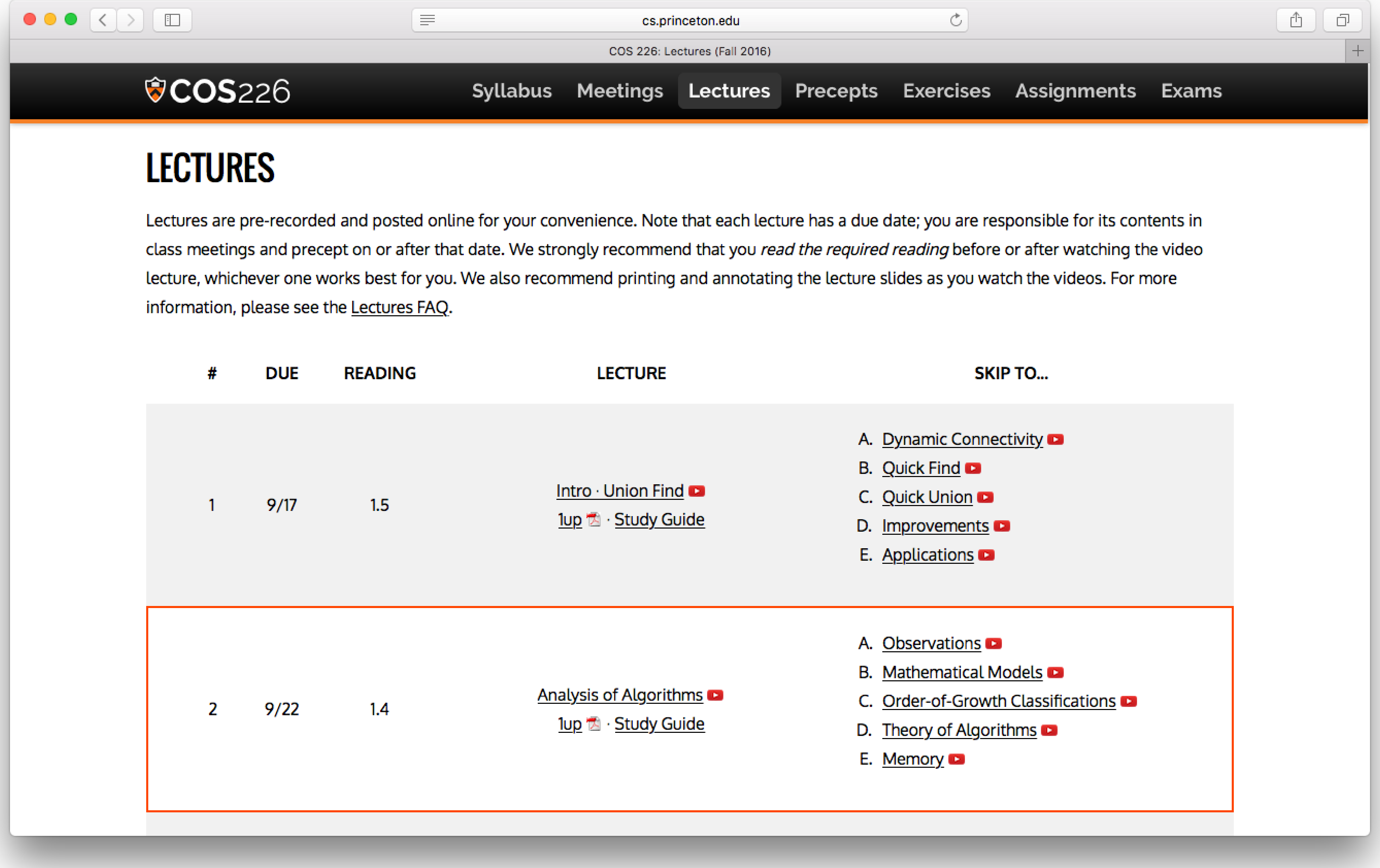
⚠ **Caution.** The assignments below have not yet been updated for this semester and may change significantly.

#	DUE	LOGO	ASSIGNMENT	COLLABORATION	SUBMIT
2	Sunday Oct. 2nd	DEQUES RANDOMIZED	**Deques & Randomized Queues** (Checklist) Implement two general-purpose data	You can work with a partner on this	Submit!

http://www.princeton.edu/~cos226 ⟵ bookmark this page!

Lectures

- Online lecture materials on course website
    - Copies of slides (.pdf and 4-up).
    - Demos
    - Study guides
    - Studio-produced videos.
    - No more live lectures.
- Approaches to reviewing a lecture
    - Watch before precept.
    - Do quizzes right away.
    - Review before exams.
    - Watch with a friend or a group



cs.princeton.edu

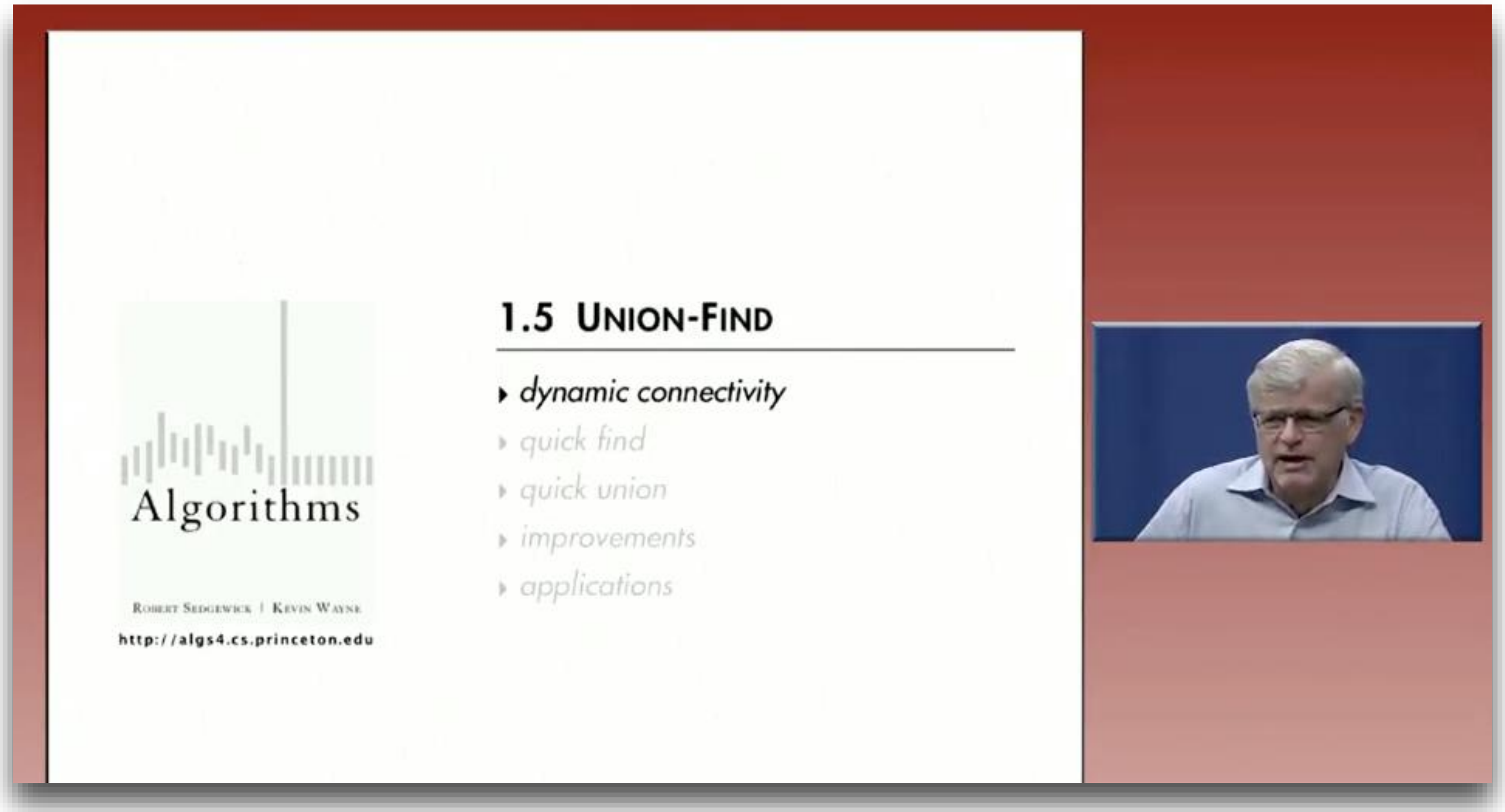
COS 226: Lectures (Fall 2016)

COS226 Syllabus Meetings Lectures Precepts Exercises Assignments Exams

LECTURES

Lectures are pre-recorded and posted online for your convenience. Note that each lecture has a due date; you are responsible for its contents in class meetings and precept on or after that date. We strongly recommend that you *read the required reading* before or after watching the video lecture, whichever one works best for you. We also recommend printing and annotating the lecture slides as you watch the videos. For more information, please see the Lectures FAQ.

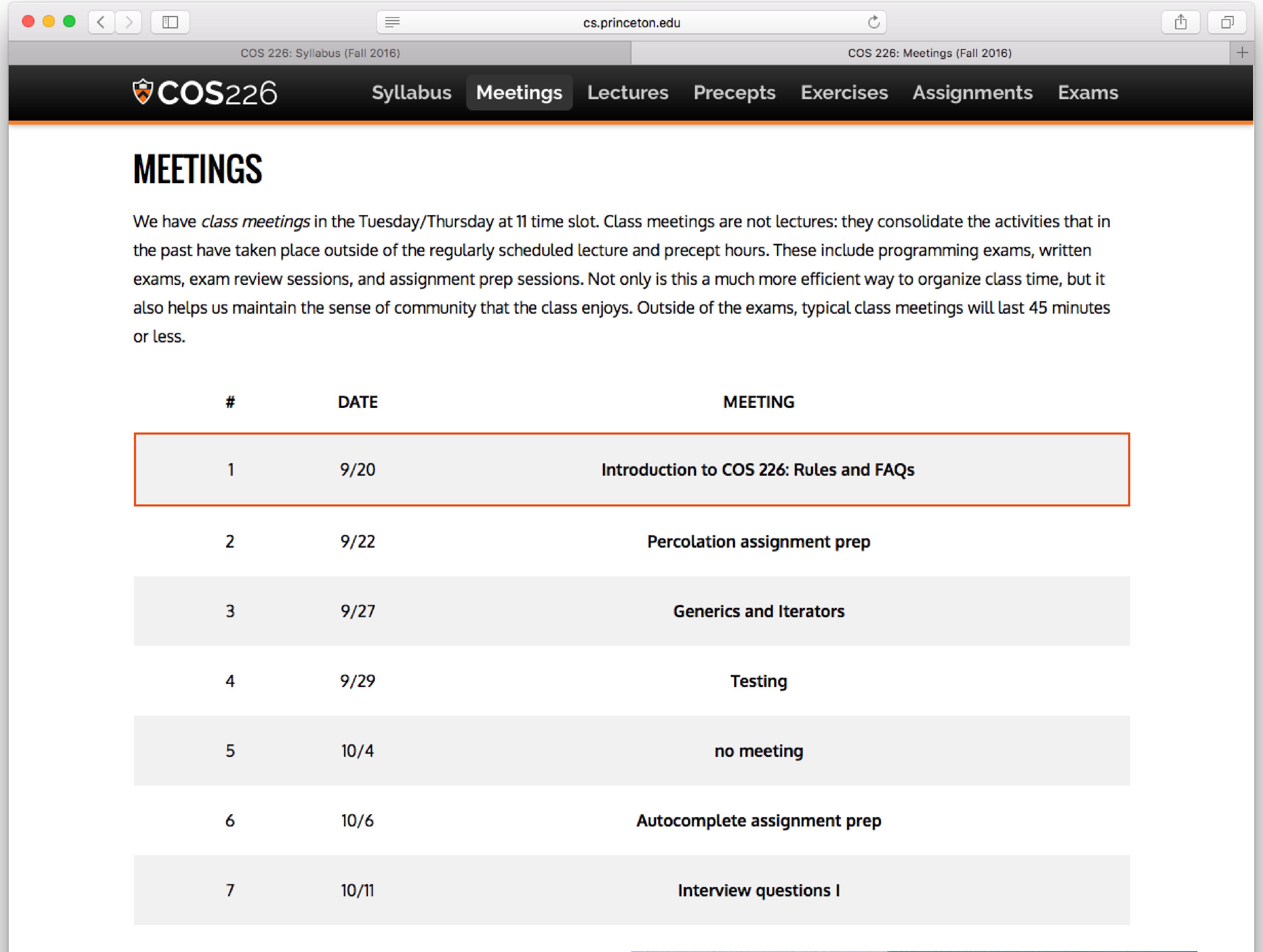
#	DUE	READING	LECTURE	SKIP TO...
1	9/17	1.5	Intro · Union Find 1up · Study Guide	A. Dynamic Connectivity B. Quick Find C. Quick Union D. Improvements E. Applications
2	9/22	1.4	Analysis of Algorithms 1up · Study Guide	A. Observations B. Mathematical Models C. Order-of-Growth Classifications D. Theory of Algorithms E. Memory



Class meetings

- We DO often meet as a class
  - Monday and Wednesday at 11.
  - Exams, review sessions, *assignment prep*.
  - Typically shorter than 80 mins (except exams).
  - Enrichment discussions to cover additional topics (more on this later)

NEXT: Percolation prep Wed. at 11.



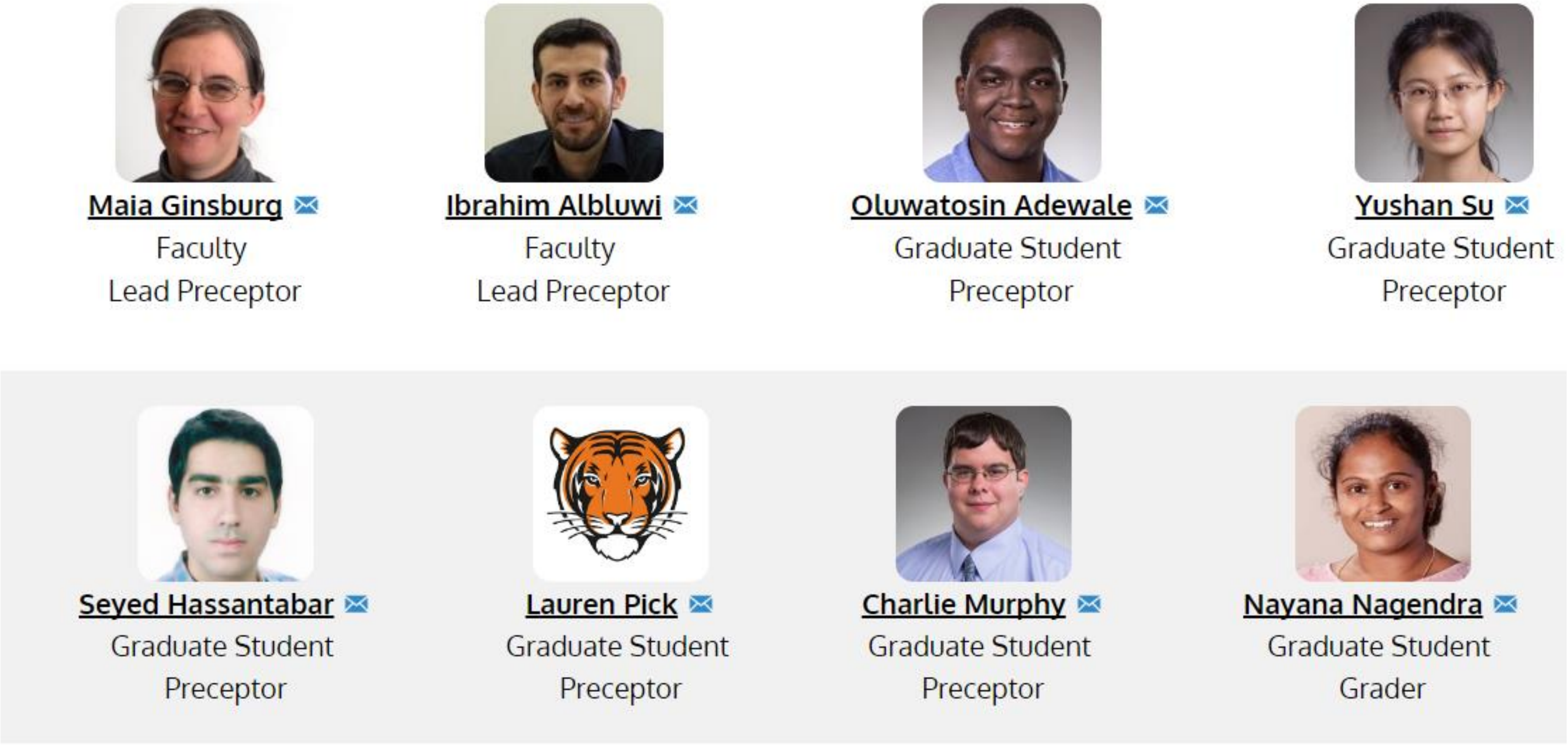
MEETINGS

We have *class meetings* in the Tuesday/Thursday at 11 time slot. Class meetings are not lectures: they consolidate the activities that in the past have taken place outside of the regularly scheduled lecture and precept hours. These include programming exams, written exams, exam review sessions, and assignment prep sessions. Not only is this a much more efficient way to organize class time, but it also helps us maintain the sense of community that the class enjoys. Outside of the exams, typical class meetings will last 45 minutes or less.

#	DATE	MEETING
1	9/20	Introduction to COS 226: Rules and FAQs
2	9/22	Percolation assignment prep
3	9/27	Generics and Iterators
4	9/29	Testing
5	10/4	no meeting
6	10/6	Autocomplete assignment prep
7	10/11	Interview questions I
8	10/13	

Precepts

- Meet once weekly.
- Get answers to questions.
- Reinforce knowledge.
- *Get to know your preceptor.*
- Attendance is 5% of your final grade (if it helps it).



Maia Ginsburg ✉
Faculty
Lead Preceptor

Ibrahim Albluwi ✉
Faculty
Lead Preceptor

Oluwatosin Adewale ✉
Graduate Student
Preceptor

Yushan Su ✉
Graduate Student
Preceptor

Seyed Hassantabar ✉
Graduate Student
Preceptor

Lauren Pick ✉
Graduate Student
Preceptor

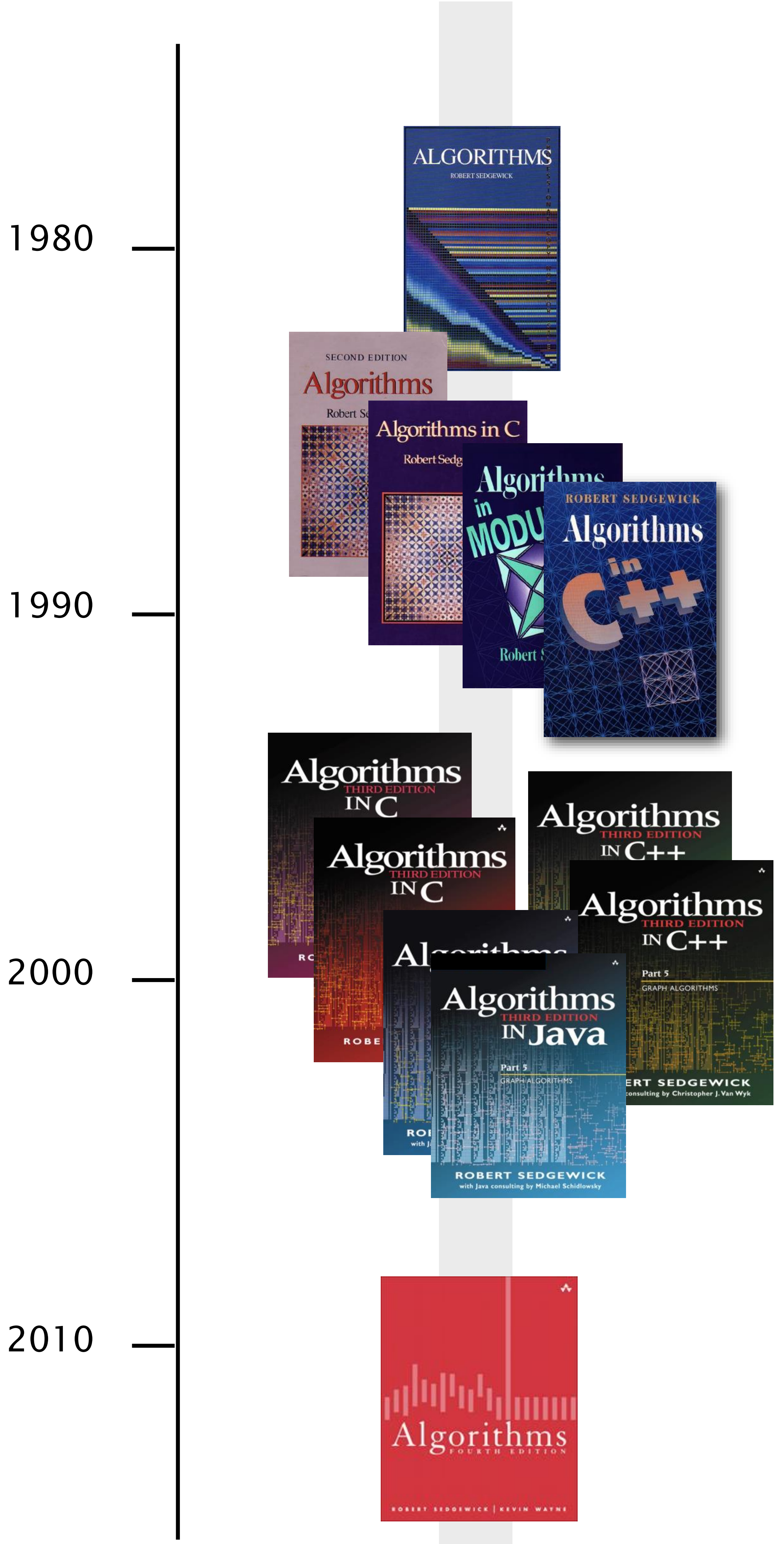
Charlie Murphy ✉
Graduate Student
Preceptor

Nayana Nagendra ✉
Graduate Student
Grader

Textbook

Algorithms, 4th edition.

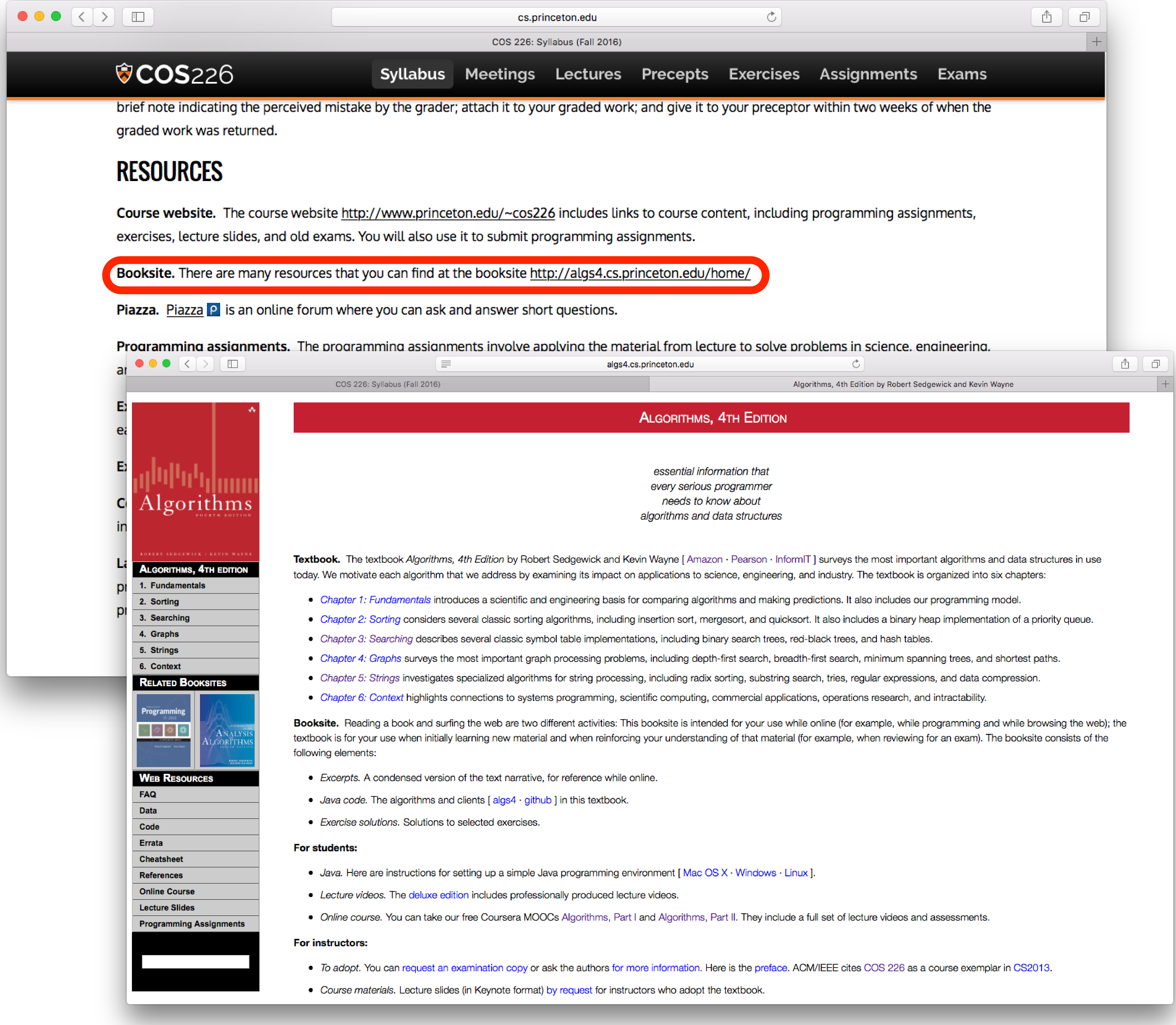
- Full coverage of course material.
- Developed for this course.
- Use while learning and studying.



Booksite

Booksite.

- Summary of content.
- Code, exercises, examples.
- Supplementary material.
- NOT the textbook.
- (also not the course web page).
- Use while online.



brief note indicating the perceived mistake by the grader; attach it to your graded work; and give it to your preceptor within two weeks of when the graded work was returned.

RESOURCES

Course website. The course website http://www.princeton.edu/~cos226 includes links to course content, including programming assignments, exercises, lecture slides, and old exams. You will also use it to submit programming assignments.

Booksite. There are many resources that you can find at the booksite http://algs4.cs.princeton.edu/home/

Piazza. Piazza is an online forum where you can ask and answer short questions.

Programming assignments. The programming assignments involve applying the material from lecture to solve problems in science, engineering,

ALGORITHMS, 4TH EDITION

essential information that
every serious programmer
needs to know about
algorithms and data structures

Textbook. The textbook *Algorithms, 4th Edition* by Robert Sedgewick and Kevin Wayne [Amazon · Pearson · InformIT] surveys the most important algorithms and data structures in use today. We motivate each algorithm that we address by examining its impact on applications to science, engineering, and industry. The textbook is organized into six chapters:

- *Chapter 1: Fundamentals* introduces a scientific and engineering basis for comparing algorithms and making predictions. It also includes our programming model.
- *Chapter 2: Sorting* considers several classic sorting algorithms, including insertion sort, mergesort, and quicksort. It also includes a binary heap implementation of a priority queue.
- *Chapter 3: Searching* describes several classic symbol table implementations, including binary search trees, red-black trees, and hash tables.
- *Chapter 4: Graphs* surveys the most important graph processing problems, including depth-first search, breadth-first search, minimum spanning trees, and shortest paths.
- *Chapter 5: Strings* investigates specialized algorithms for string processing, including radix sorting, substring search, tries, regular expressions, and data compression.
- *Chapter 6: Context* highlights connections to systems programming, scientific computing, commercial applications, operations research, and intractability.

Booksite. Reading a book and surfing the web are two different activities: This booksite is intended for your use while online (for example, while programming and while browsing the web); the textbook is for your use when initially learning new material and when reinforcing your understanding of that material (for example, when reviewing for an exam). The booksite consists of the following elements:

- *Excerpts.* A condensed version of the text narrative, for reference while online.
- *Java code.* The algorithms and clients [algs4 · github] in this textbook.
- *Exercise solutions.* Solutions to selected exercises.

For students:

- *Java.* Here are instructions for setting up a simple Java programming environment [Mac OS X · Windows · Linux].
- *Lecture videos.* The deluxe edition includes professionally produced lecture videos.
- *Online course.* You can take our free Coursera MOOCs Algorithms, Part I and Algorithms, Part II. They include a full set of lecture videos and assessments.

For instructors:

- *To adopt.* You can request an examination copy or ask the authors for more information. Here is the preface. ACM/IEEE cites COS 226 as a course exemplar in CS2013.
- *Course materials.* Lecture slides (in Keynote format) by request for instructors who adopt the textbook.

http://algs4.cs.princeton.edu

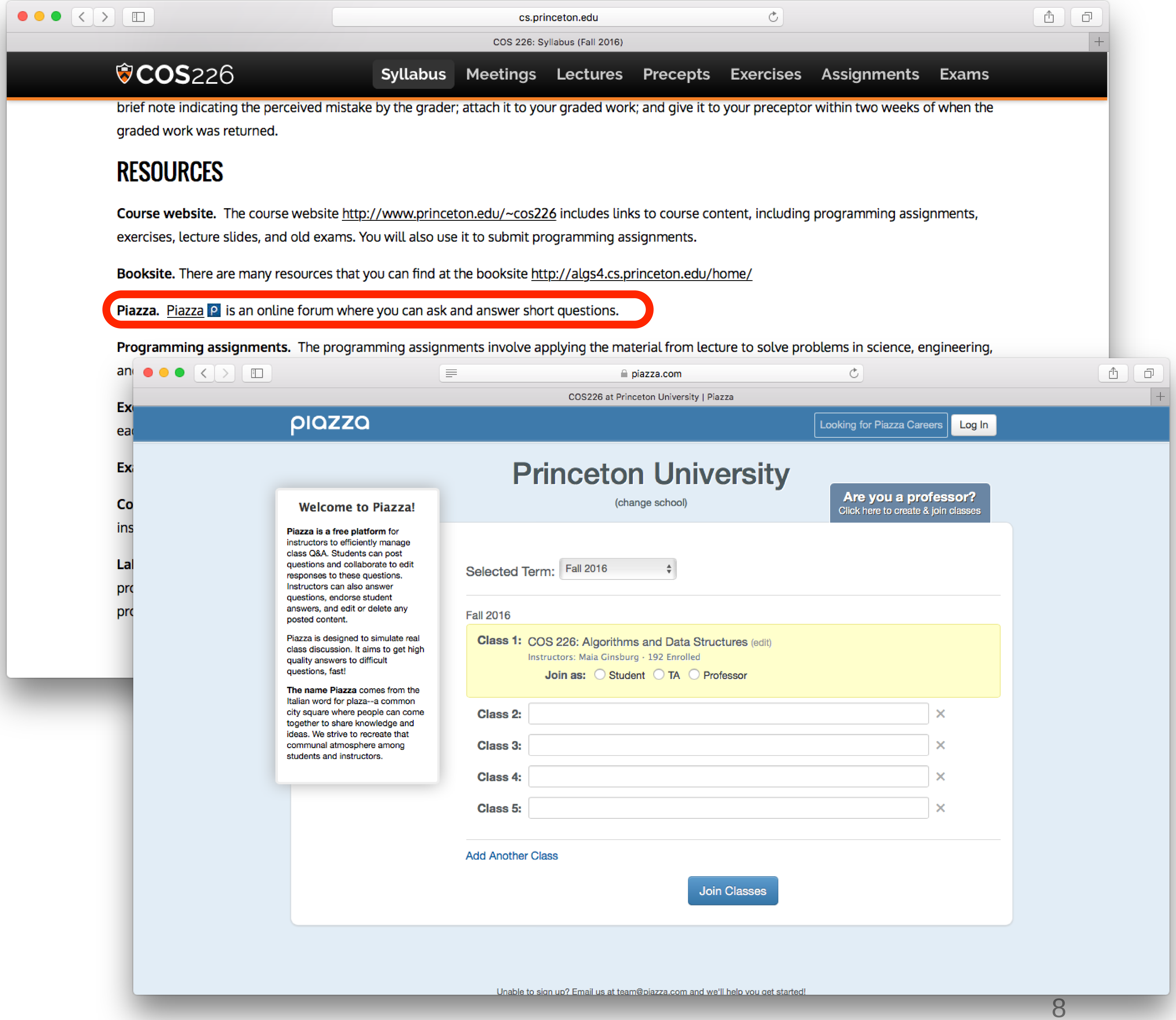
↑

bookmark this page, too!

Piazza

Piazza.

- An online forum where you can ask and answer short questions.
- More efficient and effective than asking a specific person.
- Accepted responses can improve your grade.



brief note indicating the perceived mistake by the grader; attach it to your graded work; and give it to your preceptor within two weeks of when the graded work was returned.

RESOURCES

Course website. The course website http://www.princeton.edu/~cos226 includes links to course content, including programming assignments, exercises, lecture slides, and old exams. You will also use it to submit programming assignments.

Booksite. There are many resources that you can find at the booksite http://algs4.cs.princeton.edu/home/

Piazza. Piazza P is an online forum where you can ask and answer short questions.

Programming assignments. The programming assignments involve applying the material from lecture to solve problems in science, engineering,

an

Ex
ea

Ex

Co
ins

La
pr
pr

piazza.com
COS226 at Princeton University | Piazza
piazza
Looking for Piazza Careers
Log In

Princeton University

(change school)

Are you a professor? Click here to create & join classes

Welcome to Piazza!

Piazza is a free platform for instructors to efficiently manage class Q&A. Students can post questions and collaborate to edit responses to these questions. Instructors can also answer questions, endorse student answers, and edit or delete any posted content.

Piazza is designed to simulate real class discussion. It aims to get high quality answers to difficult questions, fast!

The name Piazza comes from the Italian word for plaza--a common city square where people can come together to share knowledge and ideas. We strive to recreate that communal atmosphere among students and instructors.

Selected Term: Fall 2016

Fall 2016

Class 1: COS 226: Algorithms and Data Structures (edit)
Instructors: Maia Ginsburg · 192 Enrolled
Join as: ○ Student ○ TA ○ Professor

Class 2: ×

Class 3: ×

Class 4: ×

Class 5: ×

Add Another Class

Join Classes

Unable to sign up? Email us at team@piazza.com and we'll help you get started!

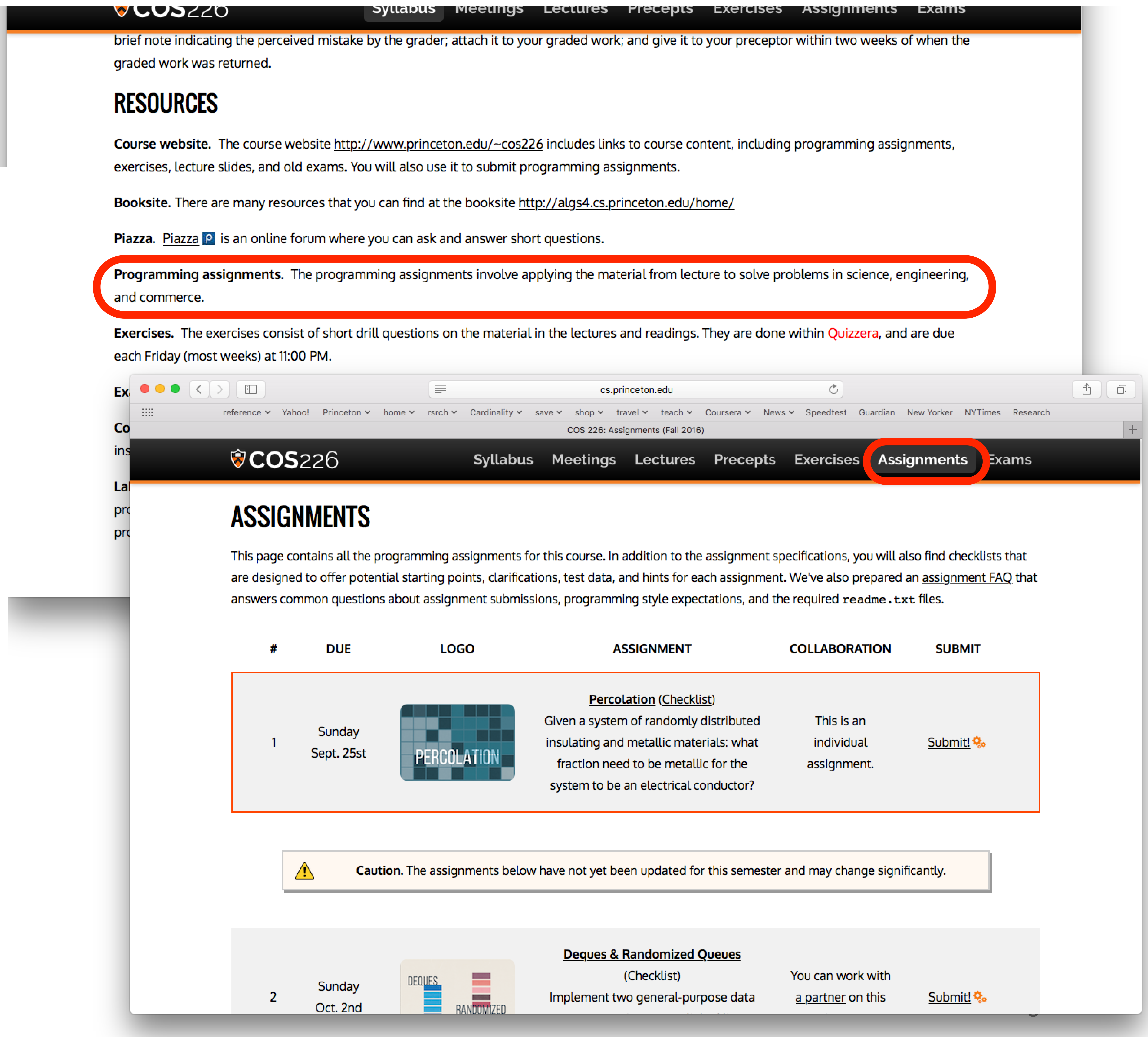
Programming assignments

Programming assignments.

- Applications of lecture material to practical problems.
- Usually due on Monday.
- **Carefully read collaboration, lateness, grading and other policies (bottom of the page).**

Special rule for first assignment

- No partnering.



brief note indicating the perceived mistake by the grader; attach it to your graded work; and give it to your preceptor within two weeks of when the graded work was returned.

RESOURCES

Course website. The course website http://www.princeton.edu/~cos226 includes links to course content, including programming assignments, exercises, lecture slides, and old exams. You will also use it to submit programming assignments.

Booksite. There are many resources that you can find at the booksite http://algs4.cs.princeton.edu/home/

Piazza. Piazza is an online forum where you can ask and answer short questions.

Programming assignments. The programming assignments involve applying the material from lecture to solve problems in science, engineering, and commerce.

Exercises. The exercises consist of short drill questions on the material in the lectures and readings. They are done within Quizzera, and are due each Friday (most weeks) at 11:00 PM.

ASSIGNMENTS

This page contains all the programming assignments for this course. In addition to the assignment specifications, you will also find checklists that are designed to offer potential starting points, clarifications, test data, and hints for each assignment. We've also prepared an assignment FAQ that answers common questions about assignment submissions, programming style expectations, and the required `readme.txt` files.

#	DUE	LOGO	ASSIGNMENT	COLLABORATION	SUBMIT
1	Sunday Sept. 25st	PERCOLATION	**Percolation** (Checklist) Given a system of randomly distributed insulating and metallic materials: what fraction need to be metallic for the system to be an electrical conductor?	This is an individual assignment.	Submit!

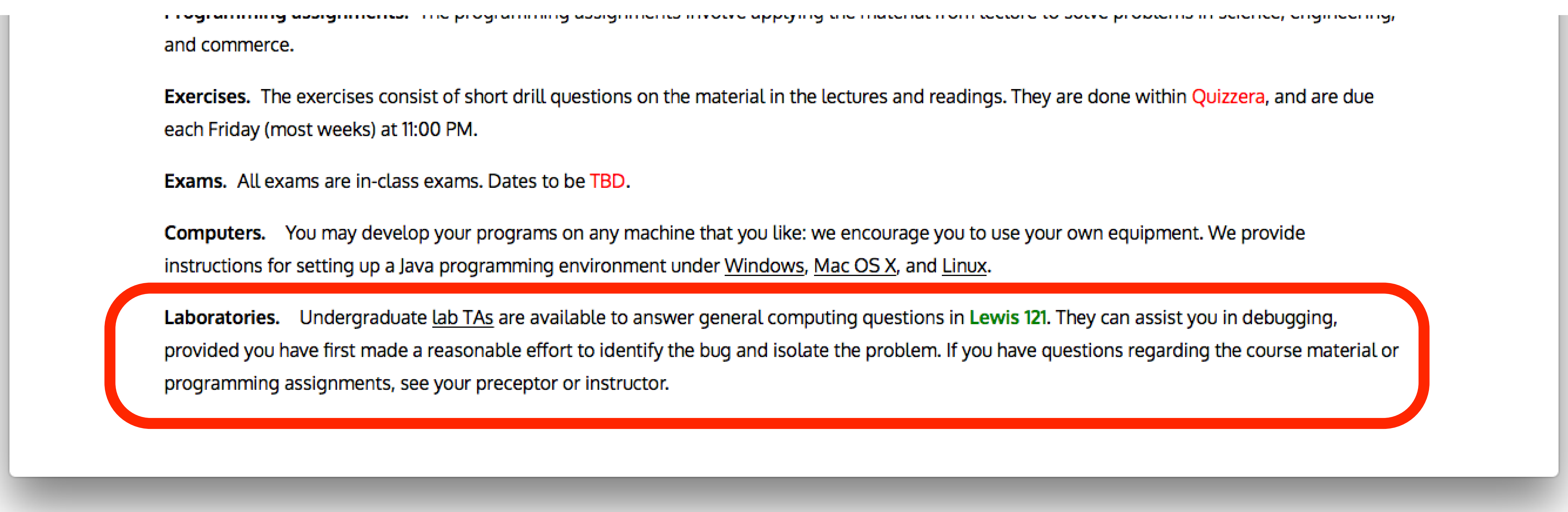
⚠ **Caution.** The assignments below have not yet been updated for this semester and may change significantly.

#	DUE	LOGO	ASSIGNMENT	COLLABORATION	SUBMIT
2	Sunday Oct. 2nd	DEQUES RANDOMIZED	**Deques & Randomized Queues** (Checklist) Implement two general-purpose data	You can work with a partner on this	Submit!

Laboratories

Lab TAs.

- Lewis 121/122.
- Help with *specific* debugging/system questions.
- Use Piazza or ask preceptor for questions on course content.



Programming assignments. The programming assignments involve applying the material from lecture to solve problems in science, engineering, and commerce.

Exercises. The exercises consist of short drill questions on the material in the lectures and readings. They are done within Quizzera, and are due each Friday (most weeks) at 11:00 PM.

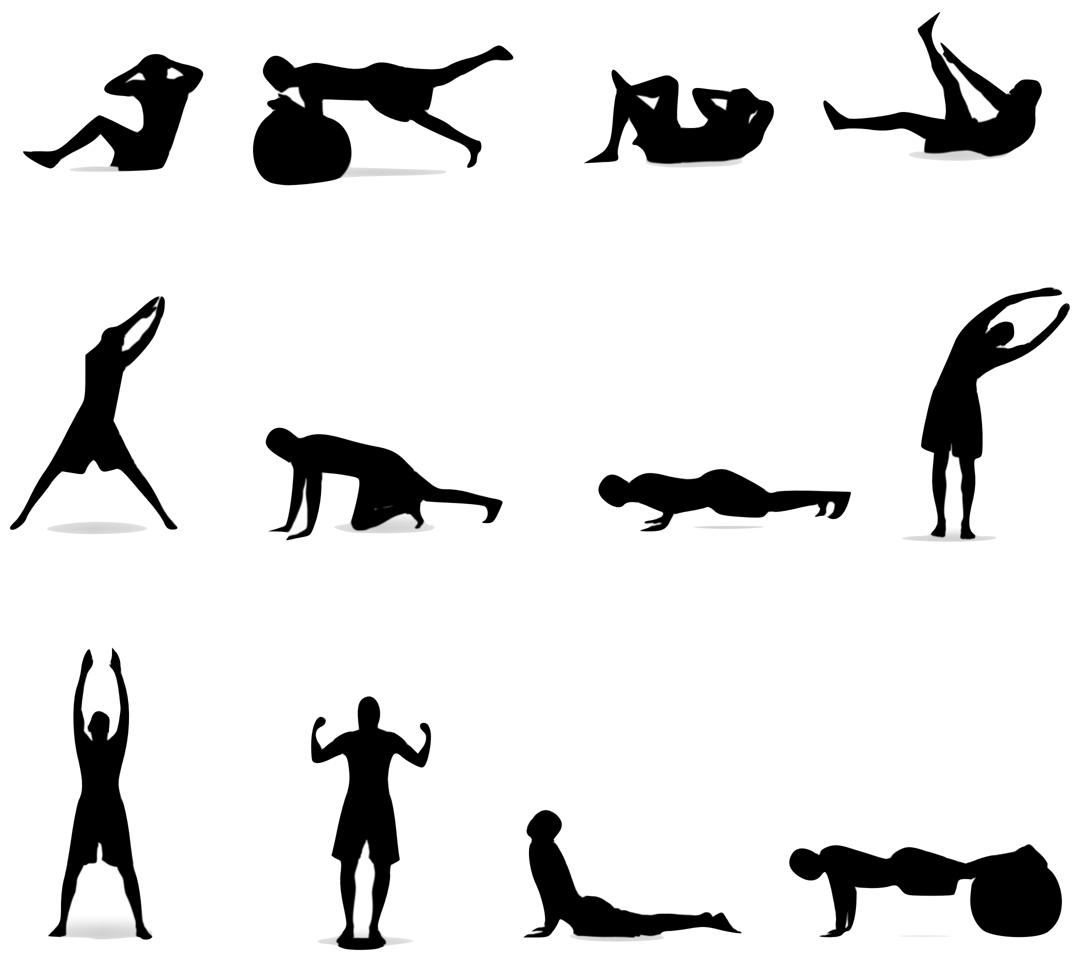
Exams. All exams are in-class exams. Dates to be TBD.

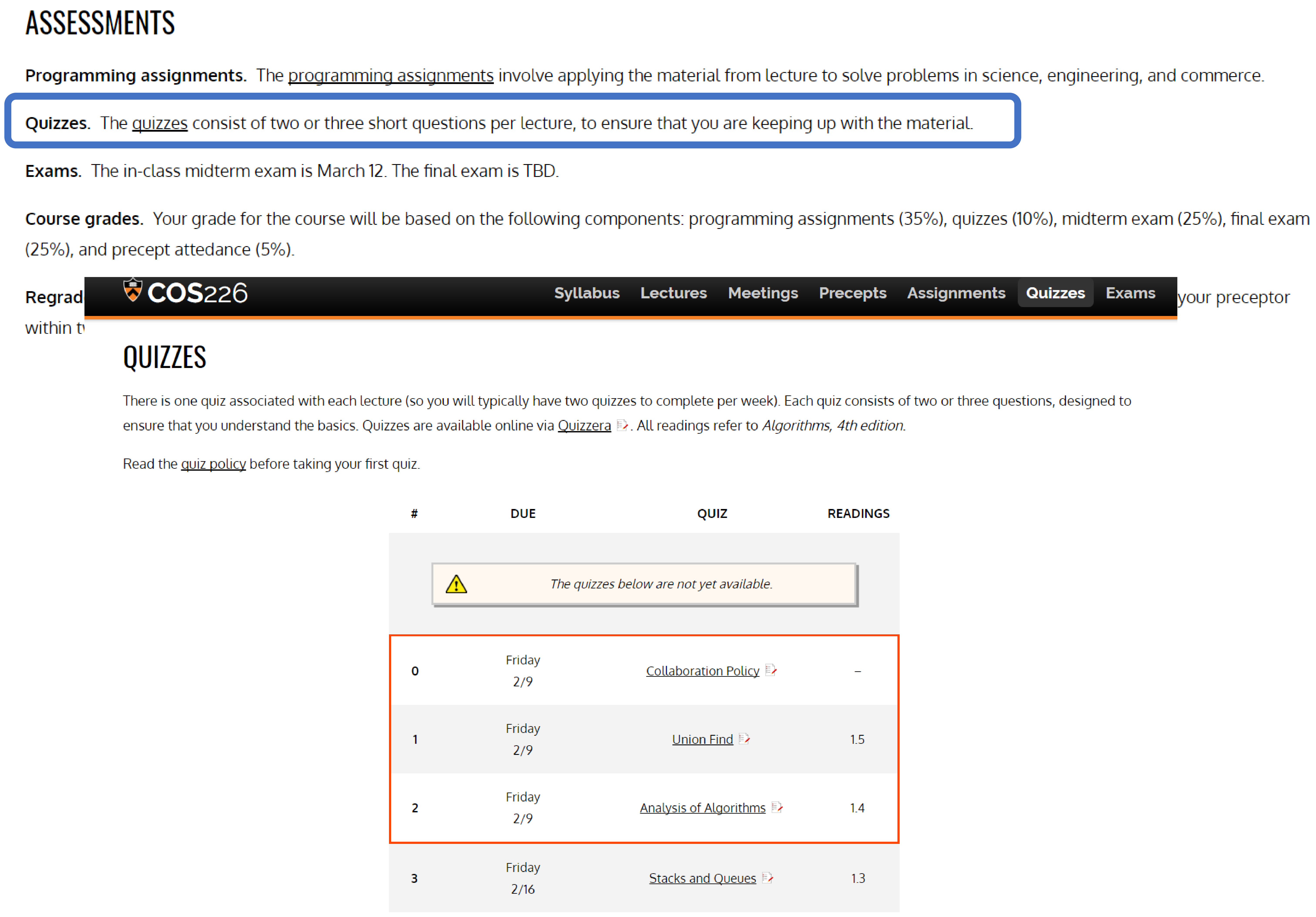
Computers. You may develop your programs on any machine that you like: we encourage you to use your own equipment. We provide instructions for setting up a Java programming environment under Windows, Mac OS X, and Linux.

Laboratories. Undergraduate lab TAs are available to answer general computing questions in Lewis 121. They can assist you in debugging, provided you have first made a reasonable effort to identify the bug and isolate the problem. If you have questions regarding the course material or programming assignments, see your preceptor or instructor.

Quizzes

- Online quizzes
  - Quick test of lecture material.
  - Usually due on Friday.
  - *Quizzera* platform developed by Princeton undergraduates.





ASSESSMENTS

Programming assignments. The programming assignments involve applying the material from lecture to solve problems in science, engineering, and commerce.

Quizzes. The quizzes consist of two or three short questions per lecture, to ensure that you are keeping up with the material.

Exams. The in-class midterm exam is March 12. The final exam is TBD.

Course grades. Your grade for the course will be based on the following components: programming assignments (35%), quizzes (10%), midterm exam (25%), final exam (25%), and precept attedance (5%).

Regrad your preceptor
within t

QUIZZES

There is one quiz associated with each lecture (so you will typically have two quizzes to complete per week). Each quiz consists of two or three questions, designed to ensure that you understand the basics. Quizzes are available online via Quizzera. All readings refer to *Algorithms, 4th edition*.

Read the quiz policy before taking your first quiz.

#	DUE	QUIZ	READINGS
⚠ *The quizzes below are not yet available.*			
0	Friday 2/9	Collaboration Policy	–
1	Friday 2/9	Union Find	1.5
2	Friday 2/9	Analysis of Algorithms	1.4
3	Friday 2/16	Stacks and Queues	1.3

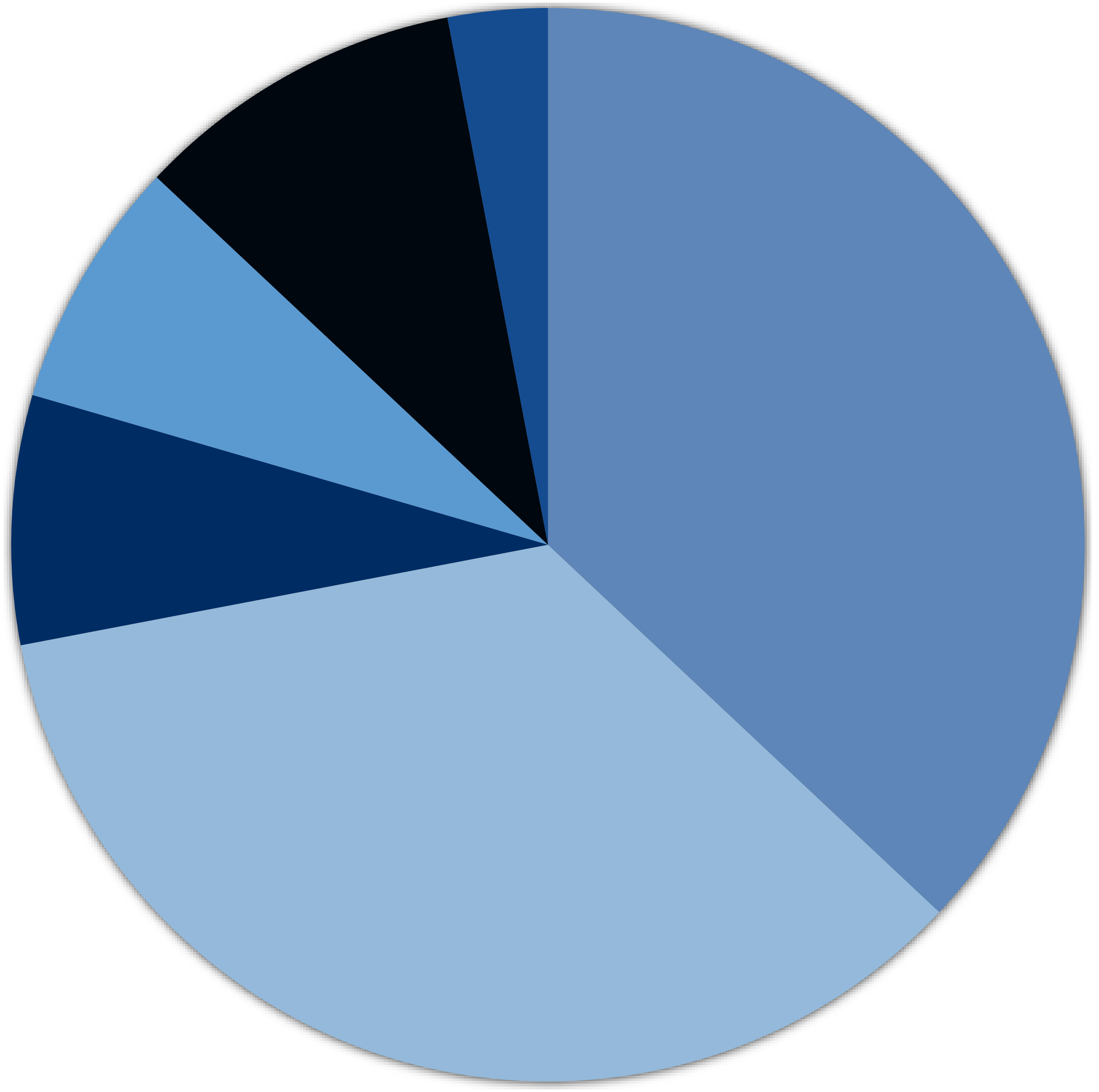
Grades

- are based on achievement.

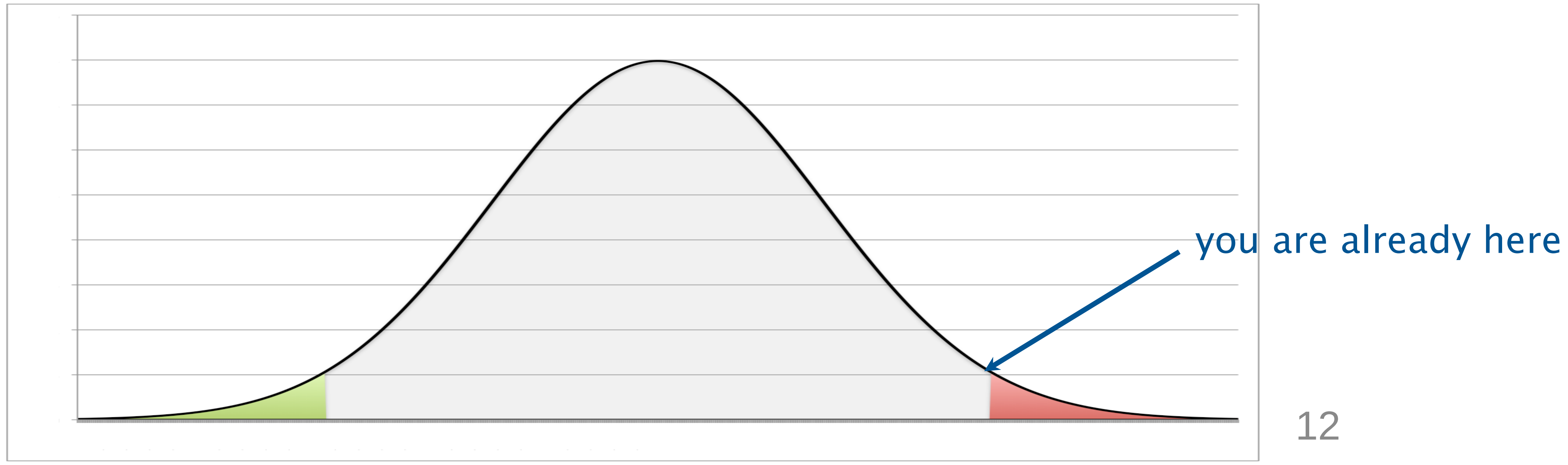
Opportunities for us to determine your level of achievement:

- 8 programming assignments.
- 2 written exams (March 12, and TBA).
- Quizzes.
- Precept attendance.
- Extra credit (including A+) / staff discretion.

Participation: Ex. Accepted Piazza responses, class meeting participation



We do not grade on a "curve".



How to do well in COS226

- Show up!

Stay engaged by:

- Doing course tasks on time: lecture, quizzes, assignments.
- Coming to class meetings and precepts.
- Preparing for precepts, and interacting with the class.
- Use resources available!

Strategies for success

Programming assignments.

- Read entire assignment carefully and write some code *before* prep session.
- Don’t get stuck on one bug.
- Use automatic checks sparingly (limit of 10, and don’t use Coursera).
- Document your code.
- **Leave time for written part.**

Exams.

- Watch lectures on time.
- Ask questions in precept.
- Don’t fall behind.

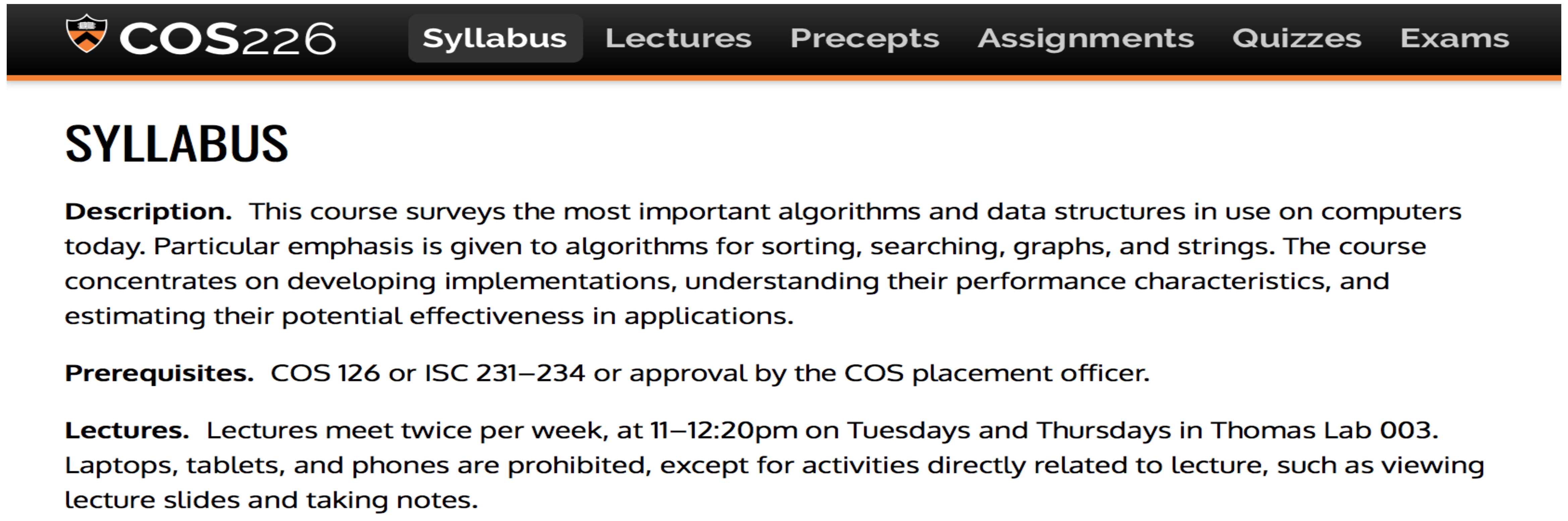
Resources (web)

Course website

- Course content.
  - Course info.
  - Lecture slides.
  - Programming assignments.
  - Quizzes.
  - Exam archive.

Booksite.

- Brief summary of content.
- Download code from book.
- APIs and Javadoc.



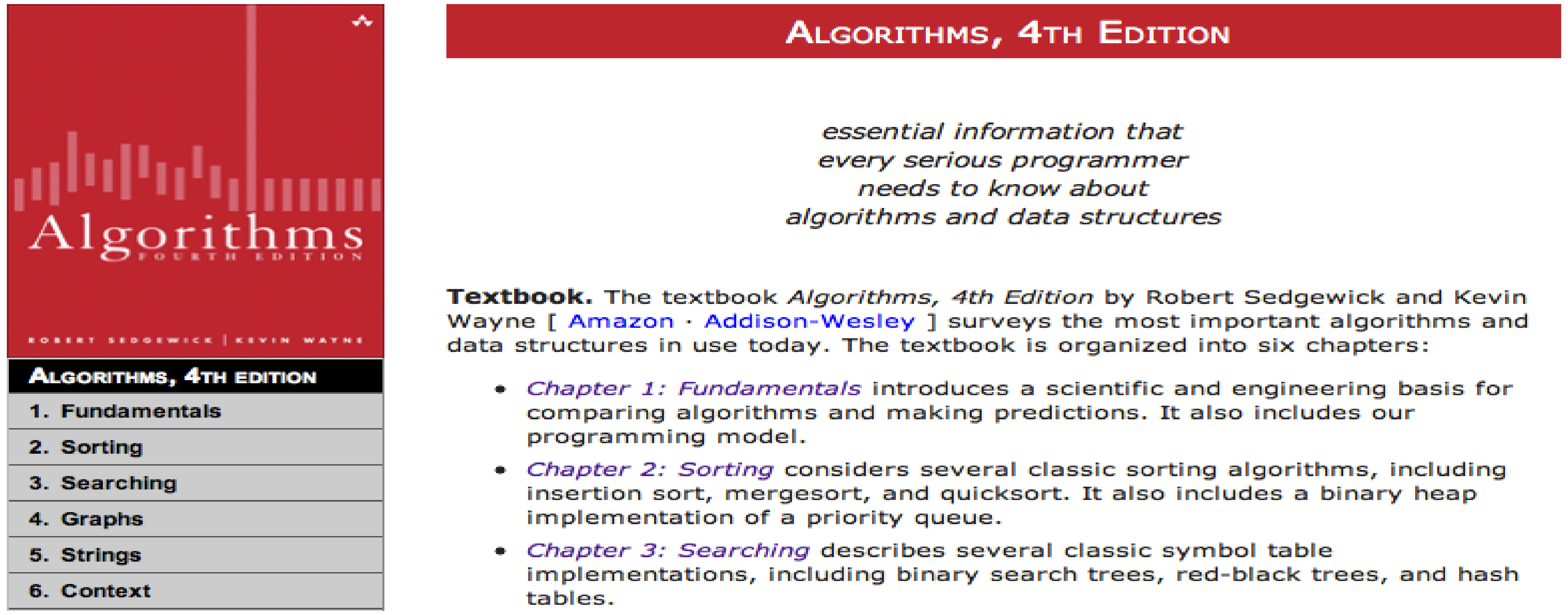
SYLLABUS

Description. This course surveys the most important algorithms and data structures in use on computers today. Particular emphasis is given to algorithms for sorting, searching, graphs, and strings. The course concentrates on developing implementations, understanding their performance characteristics, and estimating their potential effectiveness in applications.

Prerequisites. COS 126 or ISC 231–234 or approval by the COS placement officer.

Lectures. Lectures meet twice per week, at 11–12:20pm on Tuesdays and Thursdays in Thomas Lab 003. Laptops, tablets, and phones are prohibited, except for activities directly related to lecture, such as viewing lecture slides and taking notes.

http://www.princeton.edu/~cos226



ALGORITHMS, 4TH EDITION

essential information that
every serious programmer
needs to know about
algorithms and data structures

Textbook. The textbook *Algorithms, 4th Edition* by Robert Sedgewick and Kevin Wayne [Amazon · Addison-Wesley] surveys the most important algorithms and data structures in use today. The textbook is organized into six chapters:

- *Chapter 1: Fundamentals* introduces a scientific and engineering basis for comparing algorithms and making predictions. It also includes our programming model.
- *Chapter 2: Sorting* considers several classic sorting algorithms, including insertion sort, mergesort, and quicksort. It also includes a binary heap implementation of a priority queue.
- *Chapter 3: Searching* describes several classic symbol table implementations, including binary search trees, red-black trees, and hash tables.

http://algs4.cs.princeton.edu

Resources (people)

piazza

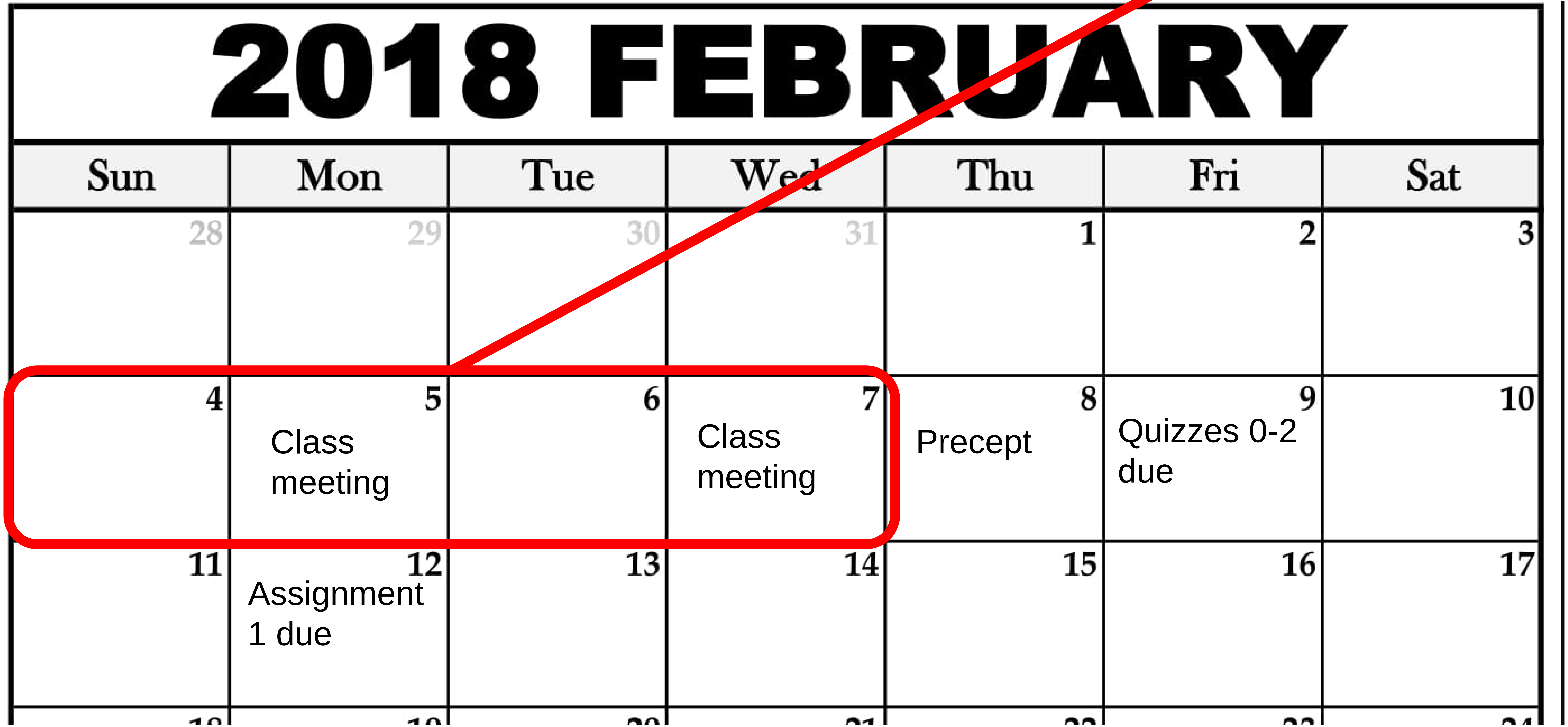
- Piazza discussion forum.
  - Low latency, low bandwidth.
  - See Piazza for guidelines.
- Office hours.
  - High bandwidth, high latency, broader than Piazza.
  - See web for schedule.
- Computing laboratory.
  - Undergrad lab TAs.
  - For help with debugging.
  - See web for schedule.

http://www.princeton.edu/~cos226

http://labta.cs.princeton.edu

A typical week in COS226

Watch first two video lecture units (Union-Find and Analysis of Algorithms)



2018 FEBRUARY

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5 Class meeting	6	7 Class meeting	8 Precept	9 Quizzes 0-2 due	10
11	12 Assignment 1 due	13	14	15	16	17
18	19	20	21	22	23	24

Q+A

- Not registered? Go to any precept this week.
- Change precept? Use TigerHub.
- All possible precepts closed? See Colleen Kenny-McGinley in CS 210.
- Haven't taken COS 126? See COS placement officer.
- Placed out of COS 126? Review Sections 1.1–1.2 of Algorithms 4/e.

Administrative portion wrap-up

- So far, discussion not really specific to Algorithms and Data Structures.

- Next: a more material-specific discussion.

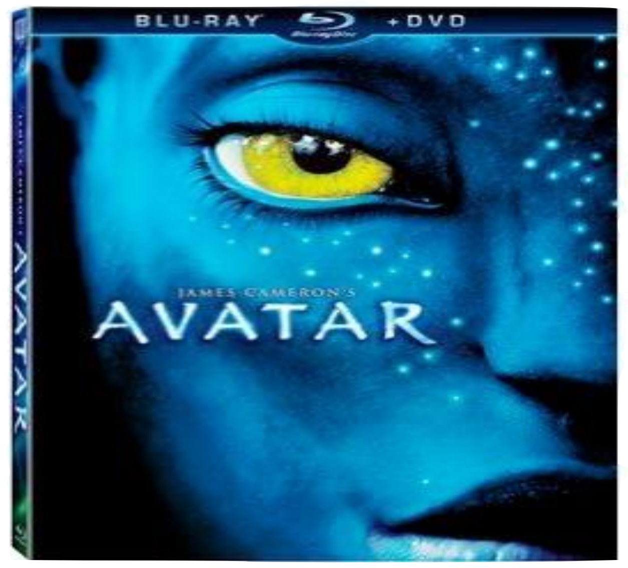
Questions and Answers: administrative part

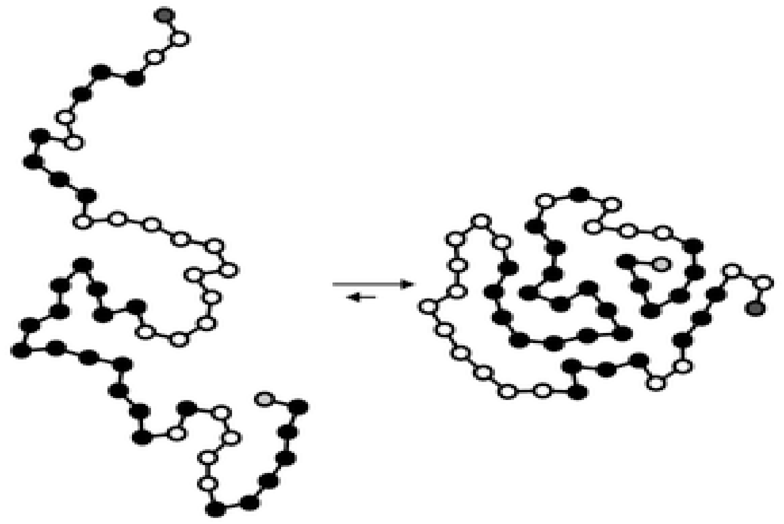
Why study algorithms and data structures?

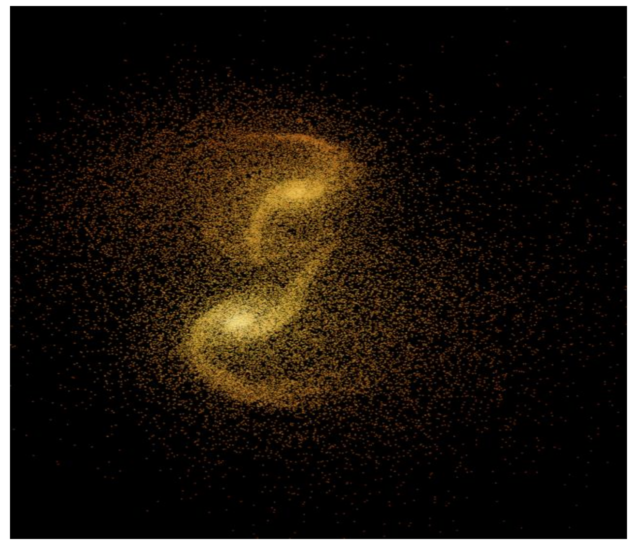
Their impact is broad and far-reaching.

- **Internet**. Web search, packet routing, distributed file sharing, ...
- **Biology**. Human genome project, protein folding, …
- **Computers**. Circuit layout, file system, compilers, …
- **Computer graphics**. Movies, video games, virtual reality, …
- **Security**. Cell phones, e-commerce, voting machines, …
- **Multimedia**. MP3, JPG, DivX, HDTV, face recognition, …
- **Social networks**. Recommendations, news feeds, advertisements, …
- **Physics**. N-body simulation, particle collision simulation, …

•

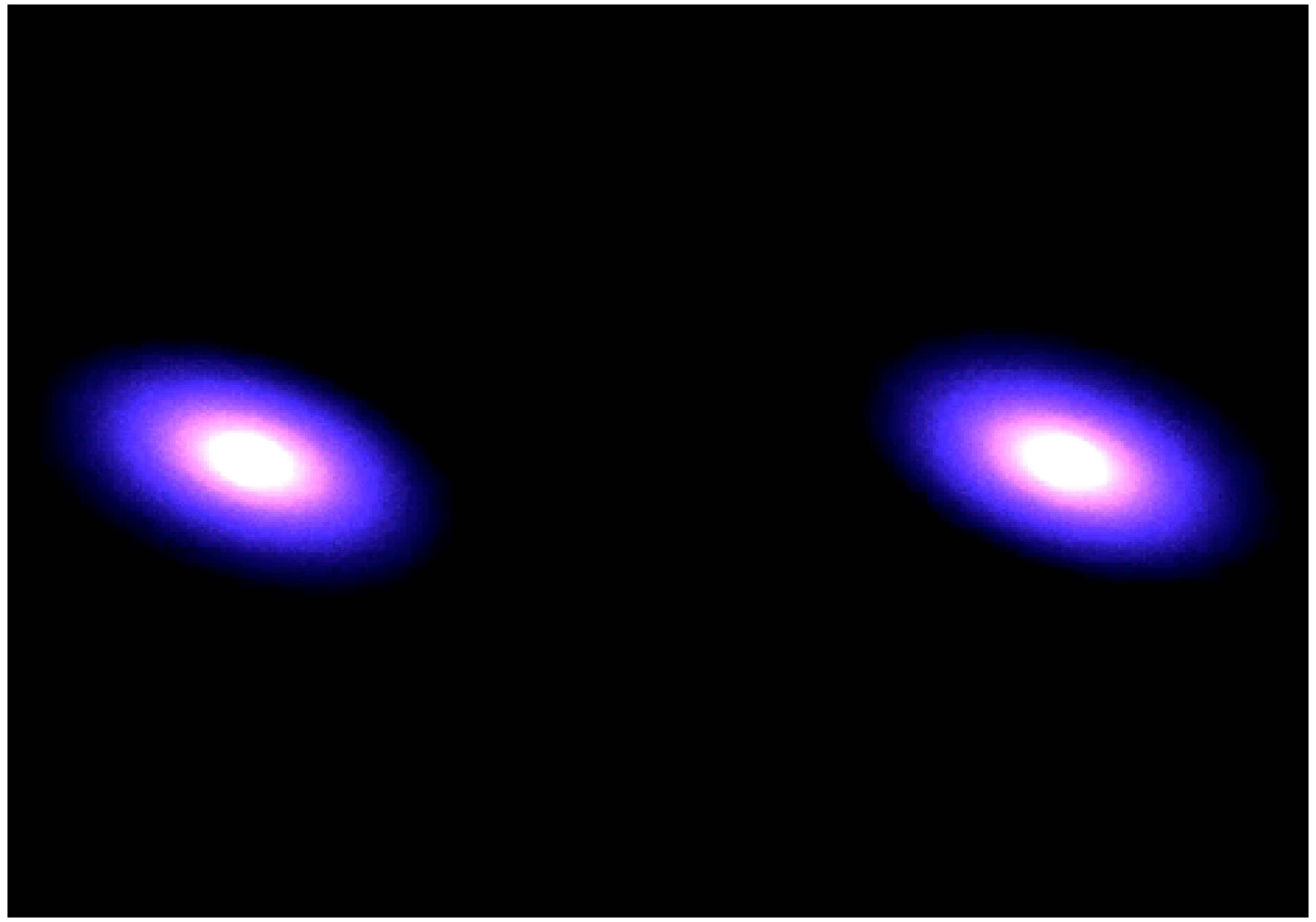






Why study algorithms and data structures?

- To solve problems that could not otherwise be addressed.



http://www.youtube.com/watch?v=ua7YlN4eL_w

Why study algorithms and data structures?

- For fun and profit.

facebook

Cisco Systems

Google

Nintendo®

JANE STREET

IBM

Adobe

Morgan Stanley

NETFLIX

RSA SECURITY™

DE Shaw & Co

ORACLE®

P PANDORA®

Akamai

YAHOO!®

amazon

Microsoft®

PIXAR ANIMATION STUDIOS

Why study algorithms and data structures?

- Significance will only increase in the near-future:
  - Moore’s law is tapering off
  - Energy draw of computing is becoming non-negligible even for the cloud
  - New applications, such as self-driving cars, internet of things, VR bring algorithmic concerns to the forefront…

Algorithms

- An algorithm is essentially a prescribed way of doing something. Typically solving a problem.
- Problems range from inspiring to mundane.
- Algorithms often use other (standard) algorithms and structures as building blocks:

```
        salary = hourlyRate * hoursWorked;
```

Could do half a
course on integer
multiplication

Data structures as building blocks

- Data structures are some of the most useful building blocks for a variety of algorithmic problems.
- A major goal of this course: to become an **empowered user** of these data structures.
- This means:
  - Being aware of their existence;
  - Understanding implementations, performance parameters and limitations;
  - Understanding the thought process that led to implementation development;
  - Being able to apply them to interesting problems;
  - Understanding what to look for when examining documentation/API for a new data structure;

Intro to algorithm design

- We will also cover some graph algorithms, which are not necessarily tied to data structures.
- These algorithms are important in their own right.
- In addition, they will teach us something about the process of designing algorithms.

Algorithms

- Typically, would like to solve a problem while conserving some resources, the relative importance of which varies from problem to problem. Examples:
  - CPU time
  - Memory
  - Latency
  - Bandwidth
  - Energy consumption
  - Reliability
  - Developer time
  - Damage due to bugs/errors

Being clever

Being wise

- An empowered user is someone who can use a combination of new but mostly off-the-shelf solutions to achieve a desired performance mix.

Summary: what the course covers

- Core:
  - Several key new data structures and algorithms
  - Experience in applying them to specific problems, and reasoning through the development process.
- In addition (these are all topics for class meetings):
  - Intro to algorithm design;
  - Furthering Java development skills;
  - Interview-style questions;
  - Additional overview topics in and around algorithms

Additional enrichment topics

- COS226 focuses on discrete, generic, algorithms and data structures, and algorithms on strings.
- Additional kinds of algorithms we will discuss in class meetings:
  - Algorithms on big integers;
  - Linear programming;
  - Continuous optimization algorithms;
- Topics around algorithms and applications:
  - Themes in algorithm design;
  - Algorithms and society, algorithmic fairness;
  - Algorithmic mechanism design;
  - Algorithmic hardness and cryptography;

Questions and Answers: course topics

Please get started on the Percolation assignment before the Wed meeting!