

# COS 333: Advanced Programming Techniques

Robert M. Dondero, Ph.D.  
Princeton University



1

Hello!  
COS 333  
Bob Dondero  
Welcome!

In today's lecture we'll cover two topics using two slide decks:

- (1) Course Overview
- (2) The Python Language (Part 1)

# COS 333

## Course Overview

Copyright © 2022 by  
Robert M. Dondero, Ph.D.  
Princeton University

2

This lecture is entitled “COS 333 Course Overview”

The goal is to give you a broad overview of the course...

So you have enough information to make a good decision about whether to take it

# Agenda

- **Introductions**
- Course Description
- Resources
- Topics
- Assignments
- Project (briefly)
- Schedule (briefly)
- Policies (briefly)
- Computing Environment

3

## **Introductions**

This is our agenda

Let's begin with introductions

## Introductions: Lead Instructor

- Bob Dondero
  - rdondero@cs.princeton.edu



4

### Introductions

#### Lead Instructor

##### School

B.A from La Salle, M.S.E. from Penn, Ph.D. from Drexel

##### Software developer

15 years, C++, Java

##### CSC teacher

La Salle, Penn State, taught in industry, Princeton

Lecturer; teaching-only faculty member

21st year

Happy to be here, looking forward to working with you!

Call me the formal “Dr. Dondero”, the informal “Bob”, or the generic “professor”.

## Introductions: TAs

- Graduate student TAs
  - One of whom will be your project adviser...

5

### **Introductions**

Graduate student teaching assistants

We have 5 graduate student teaching assistants

They will be a very important part of the course

One of them will be your project adviser

In alphabetical order by last name...

## Introductions: TAs

- Bri Butler
  - bb5943
  - @princeton.edu



## Introductions: TAs

- Anat Kleiman
  - anatk  
@princeton.edu



## Introductions: TAs

- Danqi Liao
  - dl33@princeton.edu





## Introductions: TAs

- Lisa Liu
  - xl2493
  - @princeton.edu



## Introductions: TAs

- Dora Zhao
  - dorothy  
@princeton.edu



## Introductions: Students

- Please complete **introductory survey**
  - Use **Survey App** at <https://cos333.cs.princeton.edu/Survey>
  - Your entry is due Wed 1/26 at 5:00PM
- Please sign **attendance sheet**
  - Don't sign in for another student!!!

11

### Introductions

#### Students

Please complete introductory survey

Use Survey App at <https://cos333.cs.princeton.edu/Survey>

Survey app

Goal: Inform instructors of your level of knowledge and interest  
in various course topics

I'll read your entry with great care

I'll contact you if I have any concerns

I'll tailor the course to the class's entries as much as I can

Your entry is due Wed 1/26 at 5:00PM

Please sign attendance sheet

Lecture attendance will be part of your grade, so make sure you sign  
into each lecture

Don't sign in for another student!!!

And don't ask another student to sign in for you

Last people to sign in: please give the sheets to me after class

## Agenda

- Introductions
- **Course Description**
- Resources
- Topics
- Assignments
- Project (briefly)
- Schedule (briefly)
- Policies (briefly)
- Computing Environment

# Course Description

## Goal 1: *Three-tier programming*



13

### Course Description

See the course home page or the Registrar's Office website for the official description

My view of the course is that it has 2 goals

Goal 1: three-tier programming

The course will:

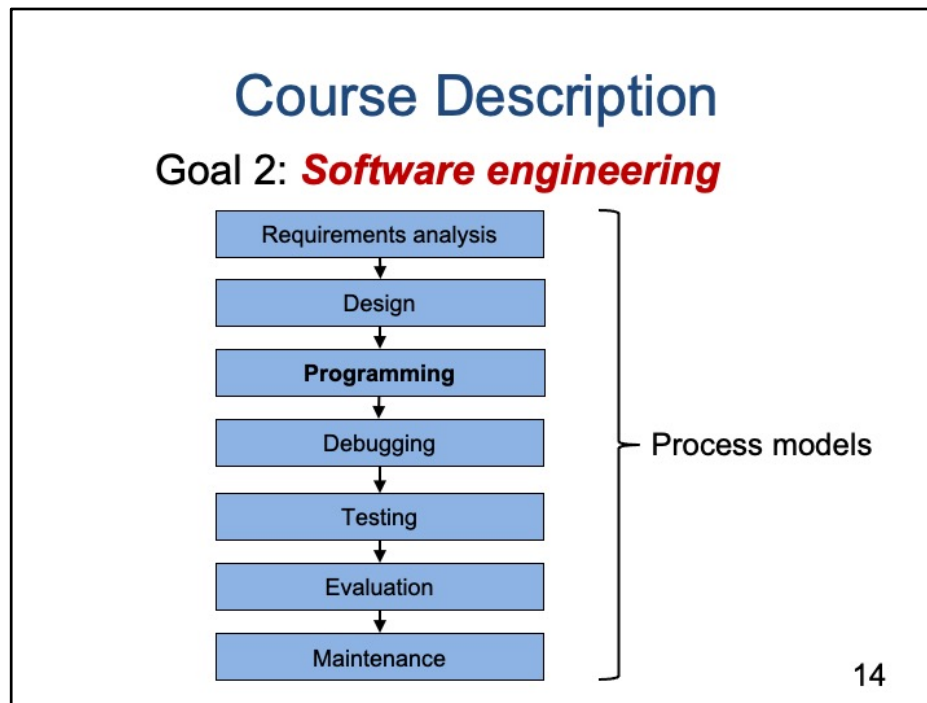
- Help you become a (better) three-tier programmer

- Help you learn how to (better) develop networked apps that consist of:

  - Data management tier: usually consists of a database

  - Presentation tier: might be a textual app, might be a graphical app, might be a web browser, or might be a native smart phone app

  - Application processing tier: the "business logic" that ties the other two tiers together



## Course Description

### Goal 2: Software engineering

The course will:

Help you become a (better) software engineer

Many CS courses at Princeton (and at other Universities) focus on programming

You're given a specification; you must compose code to implement it  
But programming is only a part of what a software engineer does

A software engineer must do requirements analyses, high-level design, programming, debugging, testing, and evaluation (which is different from testing)

A software engineer must maintain applications over time, and compose them so they are easy to maintain

A software engineer must know about process models for ordering and tracking those stages of development

This course will give you experience with those aspects of software

engineering that lie beyond programming

Some theory; lots of practice

IMHO, the most pragmatic of all Princeton COS courses

## Course Description

- Course benefits
  - Bridge the gap
  - Help you get a job!

15

### Course Description

#### Course benefits

Achieving those goals will:

Help you bridge the gap between the ACADEMIC world of  
PROGRAMMING and the REAL world of SOFTWARE  
ENGINEERING  
Help you get a job!

At least:

You'll have lots to talk about during job interviews  
You'll be able to add an impressive number of buzzwords to your  
resume



## Course Description

- How will we achieve those goals?
  - Lectures
  - Assignments
  - Project

16

### Course Description

How will we achieve those goals?

Lectures

Assignments: 5 of them

Project: all semester

## Course Description

- Course prerequisites

- COS 217
- COS 226

17

### Course Description

#### Course prerequisites

COS 217

Firm prerequisite

COS 226

Maybe concurrently, but only if you are a very good programmer (B+ or better in COS 217)

I'll check!!!

## Agenda

- Introductions
- Course Description
- **Resources**
- Topics
- Assignments
- Project (briefly)
- Schedule (briefly)
- Policies (briefly)
- Computing Environment

18

### **Resources**

Alias “sources of information”

## Resources: Website

### (1) Course website

- <https://www.cs.princeton.edu/courses/archive/spr22/cos333/index.html>

19

## Resources

### (1) Course website

We won't use Canvas

Except that you might use Canvas to access Ed

Instead we'll use an ordinary website:

<https://www.cs.princeton.edu/courses/archive/spr22/cos333/index.htm>

|

Alternative:

Browse to <http://www.cs.princeton.edu>

Click on Courses

Click on Course Schedule

Click on COS333

Make sure you visit the course website soon

## Resources: Lectures

### (2) Lectures

- Slides & code available after each lecture via *Topics* web page
- Hard copy of (mostly) code distributed during lectures
- Etiquette:
  - No electronic devices please



20

## Resources: Lectures

### (2) Lectures

[see slide]

Use of electronic devices during the lectures sometimes distracts other students  
Despite my best intentions, use of electronic devices during the lectures always  
distracts me

So reluctantly & apologetically I ask that you not use electronic devices during the  
lectures

## Resources: Ed

### (3) Ed

- Alias EdStem, Ed Discussion
- Access through Canvas:
  - <https://canvas.princeton.edu>
- Access directly:
  - <https://edstem.org/us/courses/18722/discussion/>
- Etiquette:
  - Before posting, study provided material
  - Before posting, read all (recent) Ed threads

21

## Resources

### (3) Ed

Alias EdStem, alias Ed Discussion

Access through Canvas: <https://canvas.princeton.edu>

Access directly: <https://edstem.org/us/courses/7319/discussion/>

Critical that you check often!!!

You may use for:

Issues on LECTURES

Issues on ASSIGNMENTS that are of general interest and do not  
reveal your work

Issues on your PROJECT that are of general interest

Etiquette: Before posting:

Study provided material

Read all (recent) Ed threads

## Resources: Email

### (4) Email

- To all instructors (preferred)
  - [cos333instructors@lists.cs.princeton.edu](mailto:cos333instructors@lists.cs.princeton.edu)
- To one instructor
  - See *General Information* web page or previous slides for email addresses

22

## Resources

### (4) Email

Email to all instructors:

[cos333instructors@lists.cs.princeton.edu](mailto:cos333instructors@lists.cs.princeton.edu)

Received by instructors (only)

You may use for:

Issues on LECTURES that are not of general interest

Issues on ASSIGNMENTS that are not of general interest, or that involve revealing your work

Issues on your PROJECT that are not of general interest

Email to one instructor

See General Information web page for email addresses

You may use for:

Issues that are more personal in nature

## Resources: Meetings

### (5) Instructor meetings

- See *General Information* web page for office hours

23

## Resources

### (5) Instructor meetings

See General Information web page for office hours

Current plan:

All office hours via Zoom  
Will change if appropriate

Lead instructor

Office hours

TAs

Office hours & project team meetings



## Resources: Books

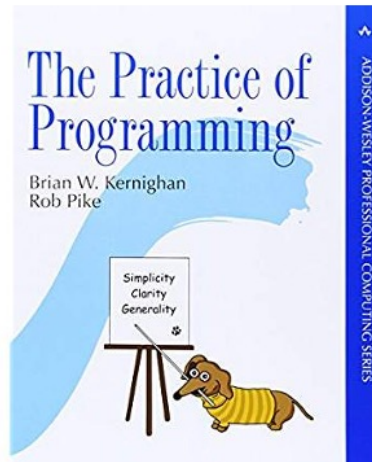
(6) Books...

24

### **Resources**

(6) Books...

## Resources: Books



Required

25

### Resources

#### Book

The Practice of Programming

Kernighan & Pike

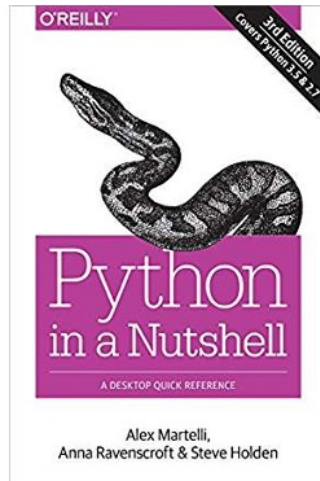
A great book, and generally will be helpful

Required, but you could survive without it

In bookstore

On reserve in Eng Library

## Resources: Books



Recommended

26

### Resources

#### Book

Python in a Nutshell (3rd edition)  
Martelli, Ravenscroft, and Holden

Recommended

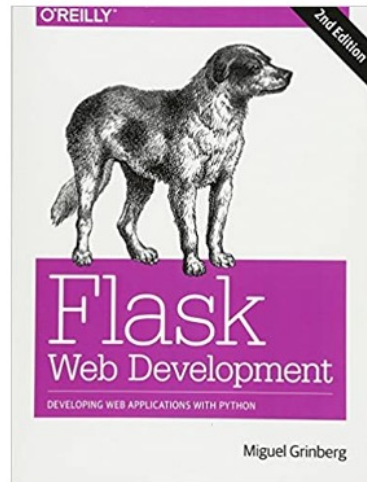
Very good REFERENCE on Python and its most fundamental  
libraries

Not a good TUTORIAL

For Python tutorials, better to use websites; course Topics page links  
to some

Available as e-book through Princeton library

## Resources: Books



Recommended

27

### Resources

#### Book

Flask Web Development (2nd edition)

Grinberg

Recommended

New to me

Seems to be a good TUTORIAL on the Flask Python server-side web framework

That topic also is covered well by some websites

Available as e-book through Princeton library

## Resources: Books



Strongly  
recommended

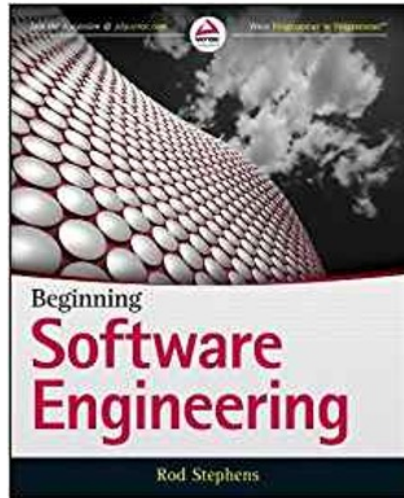
28

### Resources

#### Book

JavaScript: The Definitive Guide (7th Edition)  
Flanagan  
Strongly recommended  
Excellent tutorial on JavaScript  
Available as e-book through Princeton library

## Resources: Books



Recommended

29

### Resources

#### Book

Beginning Software Engineering  
Stephens

Good overview of the field of software engineering

Provides high-level well-balanced coverage of a wide variety of SE topics,  
with references to follow-up reading

Available as e-book through Princeton library

## Resources: Other

### (7) Other resources

- See links on *Topics* web page
- Particularly helpful:



## Resources: Summary

- Resources summary
  - (1) Course website
  - (2) Lectures
  - (3) Ed
  - (4) Email to instructors
  - (5) Meetings with instructors
  - (6) Books
  - (7) Other (e.g., Stack Overflow)



# Agenda

- Introductions
- Course Description
- Resources
- **Topics**
- Schedule (briefly)
- Assignments
- Project (briefly)
- Schedule (briefly)
- Policies (briefly)
- Computing Environment

## Topics

- See *Topics* web page
  - Subject to change...

33

### Topics

See Topics web page

Subject to change...

We will cover the following topics in the following order...

## Topics

- Version Control Systems (optional)
  - See *Version Control Systems* lecture slides
  - See *Git and GitHub Primer* document



34

### Topics

#### Version Control Systems

Specifically:

The Git version control system

The GitHub repository hosting service

Covered in COS 217

We really shouldn't cover it in COS 333

Essential for COS 333

Must use Git and GitHub for your project

Really should use them for your assignments too

Optional topic

I won't cover in lectures, but

I provided material

On course website via Topics page

Version Control Systems lecture slides

Historical overview of version control systems

Git and GitHub Primer document

Detailed instructions on how to use Git and GitHub

Please study that material if appropriate

It's entirely appropriate to ask questions about it

# Topics

- The Python Language



35

## Topics

### The Python Language

A course on programming must cover programming languages

Which came first: the thought, or the language in which the thought was expressed?!

For desktop app pgmming and server-side web pgmming, we'll use Python as our vehicle language

Soon I'll describe why

Python is not covered in any prerequisite course

So we must cover it here

We'll cover it quickly, mostly via examples

If you don't know Python, then that's perfect

If you already know Python, then this part of the course may be slow for you

Nothing I can do about it

Before you know it we'll move into topics that will be new to you

This course is not about Python

This course is about advanced programming techniques  
We'll use Python as our vehicle for studying some of those  
techniques

# Topics

- Database Programming



36

## Topics

### Database Programming

How to compose databases, and programs that use database management systems

Python will be our vehicle language

SQLite will be as our vehicle DBMS

I also will provide material on:

- PostgreSQL, as an example of a networked DBMSs

- SQLAlchemy, as an example of an object relational mapper

- MongoDB, as an example of a non-relational (NoSQL) database

Assignment 1 will be due at this point

## Topics

- Graphical User Interface (GUI) Programming



37

### Topics

#### Graphical User Interface (GUI) Programming

We'll spend most of our time studying how to develop web apps  
But we'll spend some of our time studying how to develop desktop apps...

How to compose desktop apps that have GUIs

Python will be our vehicle language  
We'll study the PyQt5 GUI package

# Topics

- Network Programming



38

## Topics

### Network Programming

How to compose programs that communicate across a network

Sockets

Communication of serialized objects across sockets

Python will be our vehicle language

Assignment 2 will be due at this point



# Topics

- Web Programming



39

## Topics

### Web Programming

The fundamentals of web programming

Hypertext transfer protocol (HTTP)

Hypertext markup language (HTML)

## Topics

- CGI Server-Side Web Programming



40

### Topics

#### CGI Server-Side Web Programming

CGI: Common Gateway Interface

The oldest and simplest way to compose web applications that generate content dynamically

Python will be our vehicle language

## Topics

- Python WSGI Server-Side Web Programming



**django**



41

## Topics

### Python WSGI Server-Side Web Programming

Another way to do server-side web programming  
Specific to Python  
Much more modern and reasonable

The Flask server-side web framework  
The Jinja template engine  
Cloud deployment to Heroku

I'll provide material on:  
The Django server-side web framework

## Topics

- Java Server-Side Web Programming



42

### Topics

#### Java Server-Side Web Programming

To complement our coverage of Python server-side web programming

Java server-side web frameworks (Spark)

Assignment 3 will be due at this point.

## Topics

- The JavaScript Language



43

## Topics

### The JavaScript Language

The second major language that we'll cover is JavaScript

Lectures will hit highlights, show examples, compare and contrast

We'll run JavaScript programs using the Node.js JavaScript runtime.

(Briefly) JavaScript server-side web frameworks

Briefly I'll show you how to do server-side web programming using JavaScript, Node.js, and Express

# Topics

- JavaScript Client-Side Web Programming



44

## Topics

### JavaScript Client-Side Web Programming

But JavaScript wasn't designed for to be run by node.js, or to do server-side web programming

Instead it was designed to be run in browsers

Running JavaScript in browsers

Asynchronous JavaScript and XML (AJAX)

JavaScript client-side web libraries (jQuery, React)

# Topics

- CSS Client-Side Web Programming



45

## Topics

### CSS Client-Side Web Programming

CSS: Cascading Style Sheets

How to compose web applications that are attractive/pretty

How to compose web applications that are responsive to the browser window size

Such applications run reasonably on a desktop computer or a smartphone

Such applications are mobile web apps

CSS Bootstrap library

Assignment 4 will be due at this point

# Topics

- Programming with Concurrent Processes



46

## Topics

### Programming with Concurrent Processes

Concurrent processes

Communication among concurrent processes

Python will be our vehicle languages



# Topics

- Programming with Concurrent Threads



47

## Topics

### Programming with Concurrent Threads

- Concurrent threads

- Thread synchronization (race conditions, deadlocks)

- Communication among concurrent threads

- Python will be our vehicle language

Assignment 5 will be due at this point

What remains are some topics that you won't need to complete the assignments, but which you might find useful to complete your project...

## Topics

- Security Issues in Web Programming



48

### Topics

#### Security Issues in Web Programming

Threats to web applications, and how to mitigate them

For example:

Thwarting injection attacks

Thwarting cross-site scripting attacks

Authentication

- Basic access authentication

- Session based authentication

- CAS

- Google authentication

Python will be our vehicle language

Flask will be our vehicle server-side framework

# Topics

- XML and JSON Programming



49

## Topics

### XML and JSON Programming

XML and JSON, mostly as data comm languages  
XML and JSON in web applications

Python and Javascript

# Topics

- Mobile Programming



50

## Topics

### Mobile Programming

- Mobile web programming
  - Already seen
- Native mobile programming
  - Java and Android
  - Swift and iOS (appendix)

# Topics

- Software engineering
  - Requirements analysis
  - Design (UML, design patterns)
  - Programming
  - Debugging
  - Testing
  - Evaluation
  - Maintenance (refactoring)
  - Process models

51

## Topics

### Software engineering

Software engineers do much more than programming

Under the heading "software engineering", we'll talk about the bigger picture

We will have been talking about software engineering – and you will have been doing software engineering -- throughout the entire course

Near the end, we'll focus on that topic

## Topics: Note

- Topics note:
  - Please contribute when you can

52

### Topics

#### Topics note

See *Topics* and *Schedule* web pages for more details

I am not an expert on all those topics!  
NOBODY could be an expert on all those topics!!!  
So please contribute when you can

Seriously!!!

In a Junior-level course, my expectation is that, on any given topic, you might know more than I do

In that case please feel free to contribute – during lectures, or via Ed

No worries if you cannot

## Agenda

- Introductions
- Course Description
- Resources
- Topics
- **Assignments**
- Project (briefly)
- Schedule (briefly)
- Policies (briefly)
- Computing Environment

## Assignments

- See *Assignments* web page
  - Hands-on experience with baseline/simple technologies
  - Teams of 2

54

### Assignments

See Assignments web page

Hands-on experience with baseline/simple technologies

Reasonably low-level technologies that you might use for your project, or that might underlie the technologies that you'll use for your project – and beyond

Teams of 2; why?

Real

The assignments are important; the project is more important



# Assignments

## • Assignment 1

- Registrar's office app: command-line version
  - Illustrates
    - Database pgmming
- **Available now**



55

## Assignments

### Assignment 1

Registrar's office app: command-line version

Princeton students want to select courses for the next semester  
You'll compose a Python Registrar's office app that allows those students to search for courses by querying a SQLite database

Textual (command-line) interface

Illustrates database programming (using indicated technologies)

Available now

# Assignments

## • Assignment 2

- Registrar's office app: desktop version 1
  - Illustrates:
    - GUI and network programming



56

## Assignments

### Assignment 2

Registrar's office app: desktop version 1

Same, but...

Illustrates GUI and network programming (using indicated technologies)

# Assignments

## • Assignment 3

- Registrar's office app: web version 1
  - Illustrates:
    - Server-side web programming



57

## Assignments

### Assignment 3

Registrar's office app: web version 1

Same, but...

Illustrates server-side web programming (using indicated technologies)

# Assignments

## • Assignment 4

- Registrar's office app: web version 2
  - Illustrates
    - Client-side web pgmming
    - Responsive web pgmming



58

## Assignments

### Assignment 4

Registrar's office app: web version 2

Same, but...

Illustrates client-side web programming & responsive web programming  
(using indicated technologies)

# Assignments

## . **Assignment 5**

- Registrar's office app: desktop version 2
  - Illustrates
    - Concurrent programming



59

## Assignments

### Assignment 5

Registrar's office app: desktop version 2

Same, but...

Illustrates concurrent programming (using indicated technologies)

## Assignments

- **Question:**
  - Why same app using different technologies?
- **Answer 1:**
  - Helps relate new technologies to old
- **Answer 2:**
  - Illustrates value of modularity

60

### Assignments

Assignments require you to implement the same app using different technologies

Question:

Why the same application using different technologies?

Answer 1:

Helps relate new technologies to old

Doing the same application using different technologies helps you to relate the new technologies to the ones that you already know

Answer 2:

The assignments are cumulative; each assignment builds upon its predecessors

The cumulative nature of the sequence illustrates the value of modularity

If your code is modular, then it will be easy to use your code from

Assignment n to complete Assignment n+1; and vice versa

# Assignments

## . **Suggestions:**

- Invest time in Assignment 1
- Choose your Assignment 1 teammate wisely

61

## **Assignments**

### Suggestions

Invest time in Assignment 1

Get the modularity right!

Generally assignments get easier as work on project ramps up

...lff you use proper modularity

Choose your Assignment 1 teammate wisely

You should have the same teammate for all assignments

Difficult to split/merge

You might ask:

“How many hours per week will you devote to the assignments?”

“When during each week will you work on the assignments?”

Looking for an Assignment 1 teammate =>

Stay after today's class

Advertise on Ed

## Agenda

- Introductions
- Course Description
- Resources
- Topics & Schedule
- Assignments
- **Project (briefly)**
- Schedule (briefly)
- Policies (briefly)
- Computing Environment



## Project

- See *Project* web page
  - Teams of 3-5
  - Networked three-tier application
  - Deliverables throughout the semester

63

### Project

See Project web page

Teams of 3-5

Networked three-tier application

Deliverables throughout the semester

More details in lecture soon

Start now; specifically...

# Project

## . *ProjectFinder App*

- <https://cos333.cs.princeton.edu/ProjectFinder>
- **Your initial entry is due Sun 1/30 at 5:00PM**

64

## Project

### ProjectFinder App

<https://cos333.cs.princeton.edu/ProjectFinder>

One entry for each student

Initially: team status, project status, technical interests, project interests

Eventually, your project name, your project description

Your initial entry is due 5:00PM Sun 1/30

You probably will revise subsequently

Goal: Help you find a project

Goal: Let others know what you're doing

## Agenda

- Introductions
- Course Description
- Resources
- Topics
- Assignments
- Project (briefly)
- **Schedule (briefly)**
- Policies (briefly)
- Computing Environment

# Schedule

See *Schedule* web page; generally...

First half	<ul style="list-style-type: none"><li>• Develop project idea</li><li>• Assemble project team</li><li>• Assignments 1, 2, 3</li><li>• Lectures related to Assignments 1, 2, 3</li><li>• Start project</li></ul>
Second half	<ul style="list-style-type: none"><li>• Assignments 4, 5</li><li>• Lectures related to Assignments 4, 5</li><li>• Lectures on more unconstrained topics</li><li>• Finish project</li></ul>

# Schedule

- Schedule notes:
  - Schedule aligns **lectures** with **assignments**
  - Schedule aligns **lectures** with **your project**?
    - Not necessarily
    - Ask me if you need lecture material ahead of pace

67

## Schedule

### Schedule notes

The schedule aligns the lectures with the assignments

The lectures will cover the material required for Assignment n just in time for you to do Assignment n

The schedule does not necessarily align the lectures with your project

How could it?

I don't know what you will be doing for your project

If, for the sake of doing your project, you need lecture material ahead of pace, then let me know

I'll do my best to get it to you ahead of pace

## Agenda

- Introductions
- Course Description
- Resources
- Topics
- Assignments
- Project (briefly)
- Schedule (briefly)
- **Policies (briefly)**
- Computing Environment

# Policies

- See *Policies* web page
  - **Grading** policies
  - **Lecture** policies
  - **Project** policies
  - **Assignment** policies

69

## Policies

### Grading policies

- Assignments (~40%)
- Project (~50%)
- Subjective (~10%)

### Lecture policies

- Lecture participation is a big part of the subjective grade
- You can participate effectively in lectures by:
  - Being there
  - Being responsive
  - Asking good questions
  - Offering good comments (not expected, but welcome)

### Project policies

- Use any resources you want
- General constraint: the work must be essentially your own
- Your TA adviser will help
- Cite your sources of info

### Assignment policies

- COS 217: closed
- COS 333: open; use any resources you want
- General constraint: the work must be essentially your own
- Specific constraint: you may not look at any COS 333 assignment solution

composed by someone else

Corollary: Make sure your assignment solutions are private, forever

Corollary: Tell me if you encounter an assignment solution  
composed by someone else

Cite your sources of info



## Agenda

- Introductions
- Course Description
- Resources
- Topics
- Assignments
- Project (briefly)
- Schedule (briefly)
- Policies (briefly)
- **Computing Environment**

## Computing Environment

- See document: *A COS 333 Computing Environment*
  - Describes how to do **assignments** on:
    - CS Department's *courselab* cluster
    - Your Mac, MS Windows computer, or Linux computer

71

### Computing Environment

See document: *A COS 333 Computing Environment*

On website via Topics page

Distributed as hard copy during first lecture

Instructions to create a computing environment to do the ASSIGNMENTS on the courselab cluster your Mac, MS Windows, or Linux computer

Poll: Who will use each kind of computer?

The document is self-explanatory

It wouldn't be appropriate – or possible – for us to cover it in lectures

Unofficial Assignment 0:

Perform the instructions in that document

Ask questions, probably on Ed, if you have any trouble

In closing...

## Action Items

- By Wed 1/26 5:00PM
  - Use **Survey App** to express your expertise and interest in course topics
    - <https://cos333.cs.princeton.edu/Survey>

## Action Items

- By Sun 1/30 5:00PM
  - Use ***ProjectFinder App*** to indicate your project status and interests
    - <https://cos333.cs.princeton.edu/ProjectFinder>

## Action Items

- Soon
  - Read **course website**, esp. *Policies* page
    - <https://www.cs.princeton.edu/courses/archive/spr22/cos333/index.html>
  - Make sure you're comfortable with Git and GitHub
    - Version Control Systems lecture slides
    - ***Git and GitHub Primer*** doc
  - Create a COS 333 computing env
    - ***A COS 333 Computing Env*** doc

# Summary

- Course overview
  - Introductions
  - Course Description
  - Resources
  - Topics
  - Assignments
  - Project (briefly)
  - Schedule (briefly)
  - Policies (briefly)
  - Computing Environment