Here Early?

Here for CSE 311?

Welcome! You’re early!

Want a copy of these slides to take notes?

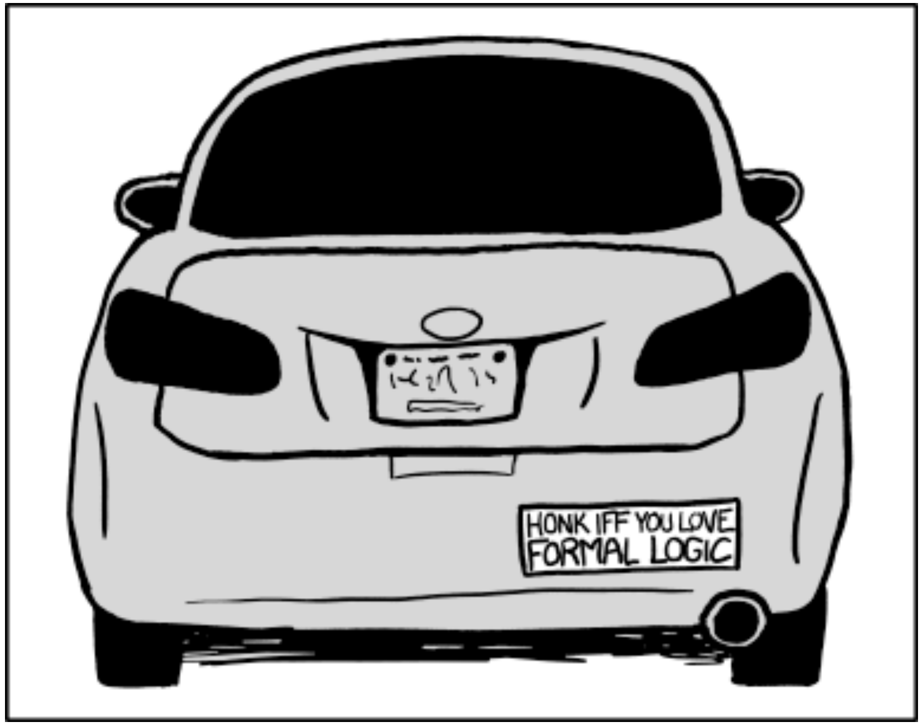
You can download them from the webpage cs.uw.edu/311

Want to be ready for the end of the lecture?

Download the Activity slide from the same place

Go to pollev.com/cse311 and login with your at-uw email (not at-cs!)

You should see a multiple choice question



Logistics and Propositional Logic

CSE 311: Foundations of Computing I
Lecture 1

Outline

Course logistics (e.g. how are we doing this online?)

What is the goal of this course?

Start of Propositional Logic

Zoom Logistics

We’ll always have a TA watching chat – if you have a question:

- Preferably, please unmute yourself and ask!
- Or, ask it in the chat (either general or direct to the TA).

    Don’t send direct to me, I won’t see it 😬

TA may answer directly, interrupt me, or wait a few minutes and have me answer at a good stopping point.

If you’re comfortable (and have the wifi) to turn on your video please do

Nodding/confused looks/glazed over eyes help me know if I said something super confusing.

We will put recordings of (both) lectures on the course webpage.

Staff

Instructor: Jamie Morgenstern

Email: jamiemmt@cs.washington.edu

TAs

Sandy	Chien
Daniel Thomas	Fuchs
Philip	Garrison
Sangwon	Kim
Andrey	Ryabtsev
Alicia Maximovna	Stepin
Ivy	Wang
Alice	Wang

Sections

Sections start tomorrow!

Mostly a chance to practice and ask questions

Please attend your registered section if you can.

There can be multiple sections at the same time, make sure you know the two-letter code for your section.

Zoom links on Canvas or Ed.

Some sections introduce new material.

TA walkthroughs will be posted for reference, but sections aren't recorded.

Syllabus

It's all on the webpage:
https://courses.cs.washington.edu/courses/cse311/21wi

In general, when in doubt, it's on the webpage.

We'll talk through syllabus details as they become relevant, only a few highlights today…

Textbook

We'll have occasional pre- or post-lecture readings. All required readings will be available on the webpage.

There is also an **optional** Book:

Discrete Mathematics and its Applications (Kenneth Rosen)

We'll tell you the relevant sections for 6^{th} or 7^{th} editions.

- Many used copies available
- Good for practice with solved problems

Older (or newer) editions also have necessary content, but it may be moved around.

Work

Homework (70%)

Approximately weekly. Mostly due Fridays.

Graded on both accuracy and clarity/style.

Exams (22.5%)

We’ll have two take-home exams (think “shorter homework” rather than one-hour exam). Approximate dates: Nov. 13-16, Dec. 11-14

Lecture activities (7.5%)

Completed either online “live” or (if you’re asynchronous, or miss a lecture) online by the following Sunday.

Communication

Ed Discussion board will be our primary means of communication.

Please check frequently.

You are also already be on the class email list

Occasional announcements here.

If you want to contact us:

- Private post on Ed (seen by staff, all TAs)
- Email cse311-staff@cs.washington.edu
- Anonymous Feedback form on webpage

Pre-Quarter Survey

There's a "quiz" up on canvas.

Asking you questions like "what time zone are you in?"

This will help us schedule office hours, connect people who might want study groups, etc.

Please fill it out by tonight!

Collaboration Policy

PLEASE collaborate! Please talk to each other and work with each other.

(subject to the policy – details on webpage)

We're remote – it's going to be harder to find people to work with.

- Ed posts to help find people
- Stay after section tomorrow
- Pre-course survey to help asynchronous people
- Let us know how we can help.

Form Study Groups!

311 is just a *different* course than intro programming.

If programming “came easy” for you, 311 might not (and vice versa).

Form a study group!

"when people said form study groups they meant form study groups“
--Chloe Dolese Mandeville, CSE advisor

CSE 390Z

CSE 390Z is a workshop designed to provide academic support to students enrolled concurrently in CSE 311. During each 1.5-hour workshop, students will reinforce concepts through:

- **collaborative problem solving**
- **practice study skills and effective learning habits**
- **build community for peer support**

All students enrolled in CSE 311 are welcome to register for this class.If you are interested in receiving an add code, please fill out a form here. If you have any questions or concerns please contact Rob (minneker@uw.edu).

We’re in a pandemic…

…this just isn’t normal.

Please, contact us *early* and often any time you might have anything challenging come up.

and this is usually easier the earlier we know about an issue.

We’re going to do our best to support you

If there’s something you’re “missing” that we can help with tell us!

What is this course?

What is this course?

In this course, you will learn how to make and communicate rigorous and formal arguments.

Why? Because you'll have to do technical communication in real life.

If you become a PM – you'll have to convert non-technical requirements from experts into clear, unambiguous statements of what is needed.

If you become an engineer – you'll have to justify to others exactly why your code works, and interpret precise requirements from your PM.

If you become an academic – to explain to other academics how your algorithms and ideas improve on everyone else's.

What is this course?

In this course, you will learn how to make and communicate rigorous and formal arguments.

Two verbs

Make arguments – what kind of reasoning is allowed and what kind of reasoning can lead to errors?

Communicate arguments – using one of the common languages of computer scientists (no one is going to use your code if you can't tell them what it does or convince them it's functional)

Course Outline

Symbolic Logic (training wheels; lectures 1-8)

Just make arguments in mechanical ways.

- Using notation and rules a computer could understand.

Understand the rules that are allowed, without worrying about pretty words.

Set Theory/Arithmetic (bike in your backyard; lectures 9-20)

Make arguments, and communicate them to humans

Arguments about numbers and sets, objects you already know

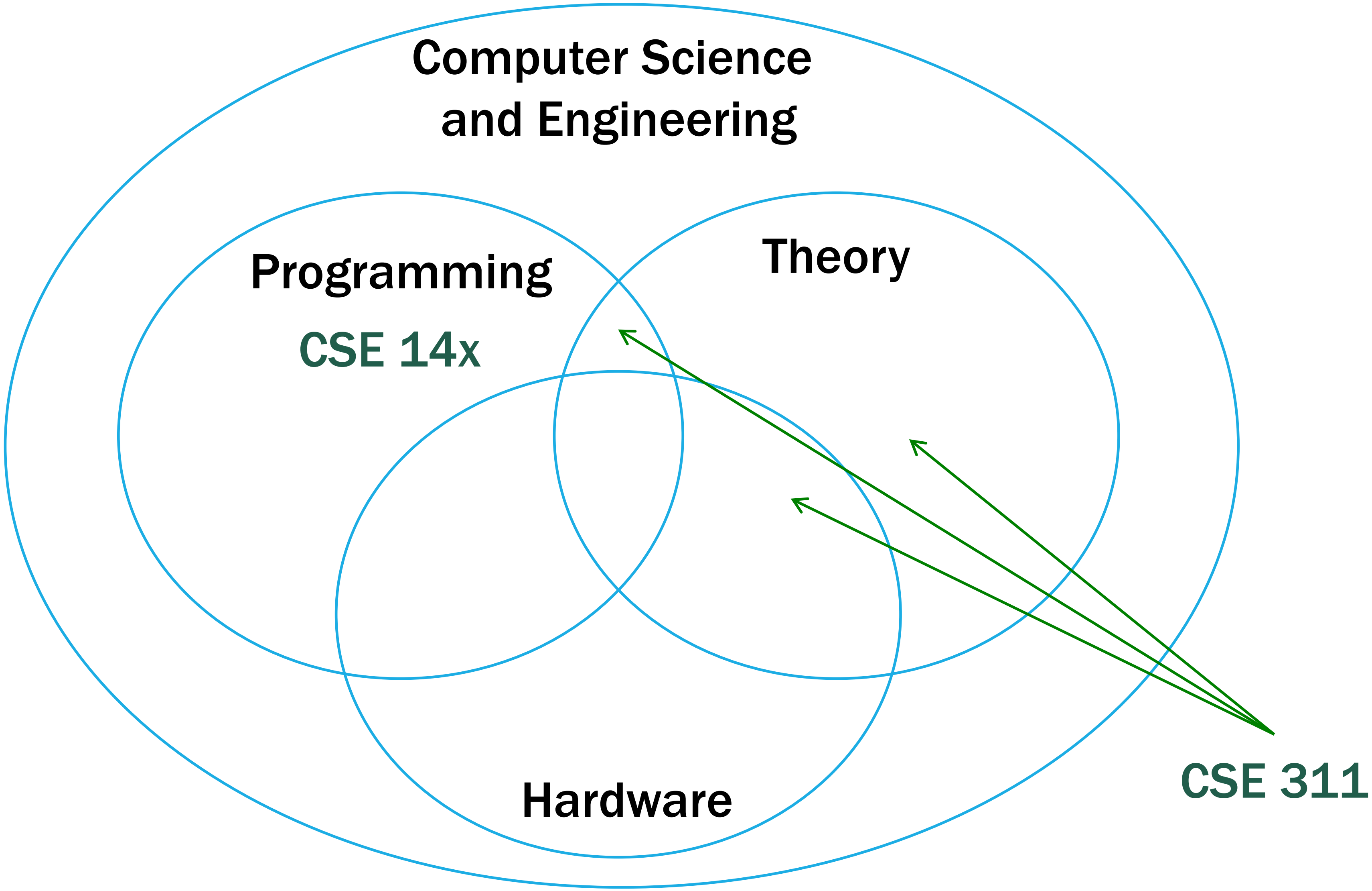
Models of computation (biking in your neighborhood; lectures 21-30)

Still make and communicate rigorous arguments

But now with objects you haven't used before.

- A first taste of how we can argue rigorously about computers.

Some Perspective



Symbolic Logic

What is symbolic logic and why do we need it?

Symbolic Logic is a language, like English or Java, with its own

words and rules for combining words into sentences (syntax)

ways to assign meaning to words and sentences (semantics)

Symbolic Logic will let us **mechanically** simplify expressions and make arguments.

The new language will let us focus on the (sometimes familiar, sometimes unfamiliar) rules of logic.

Once we have those rules down, we'll be able to apply them "intuitively" and won't need the symbolic representation as often

but we'll still go back to it when things get complicated.

Propositions: building blocks of logic

Proposition

A statement that has a truth value (i.e. is true or false) and is "well-formed"

Propositions are the basic building blocks in symbolic logic.

Here are two propositions.

All cats are mammals

True, (and a proposition)

All mammals are cats

False, but is well-formed and has a truth value, so still a proposition.

Analogy

In 142/143 you talked about a variable type that could be either true or false.

You called it a “`Boolean`”

Boolean variables are a useful analogy for propositions.

- They aren’t identical, but they’re very similar.

Are These Propositions?

$2 + 2 = 5$ This is a proposition. It's okay for propositions to be false.

$x + 2 = 5$ Not a proposition. Doesn't have a fixed truth value

Akjsdf! Not a proposition because it's gibberish.

Who are you?

This is a question which means it doesn't have a truth value.

There is life on Mars.

This is a proposition. We don't know if it's true or false, but we know it's one of them!

Propositions

We need a way of talking about *arbitrary* ideas…

To make statements easier to read we'll use propositional variables like $p, q, r, s, \ldots$

Lower-case letters are standard.

Usually start with p (for proposition), and avoid t, f, because…

Truth Values:

T for true (note capitalization)

F for false

Analogy

We said propositions were a lot like Booleans…

How did you connect Booleans in code?

```
&&
||
!
```

Logical Connectives

And (&&) works exactly like it did in code.

But with a different symbol ∧

Or (||) works exactly like it did in code.

But with a different symbol ∨

Not (!)works exactly like it did in code.

But with a different symbol ¬

Some Truth Tables

p	$\neg p$

p	q	$p \wedge q$

p	q	$p \vee q$

Truth tables are the simplest way to describe how logical connectives operate.

Some Truth Tables

p	$\neg p$
T	F
F	T

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Truth tables are the simplest way to describe how logical connectives operate.

Implication

Another way to connect propositions

If p then q.

"If it is raining, then I have my umbrella."

$p \to q$

Think of an implication as a promise.

Implication

The first two lines should match your intuition.

The last two lines are called “vacuous truth.” For now, they’re the definition. We’ll explain why in a few lectures.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

This is the definition of implication. When you write “if…then…” in a piece of mathematical English, this is how you will be interpreted.

Implication ($p \to q$)

"If it's raining, then I have my umbrella"

It's useful to think of implications as promises. An implication is false exactly when you can ***demonstrate*** *I'm lying.*

	It's raining	It's not raining
I have my umbrella		
I do not have my umbrella		

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Implication ($p \rightarrow q$)

"If it's raining, then I have my umbrella"

It's useful to think of implications as promises. An implication is false exactly when you can ***demonstrate*** *I'm lying.*

	It's raining	It's not raining
I have my umbrella	No lie. True	No lie. True
I do not have my umbrella	**LIE!** **False**	No lie. True

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

$p \rightarrow q$

$p \rightarrow q$ and $q \rightarrow p$ are different implications!

"If the sun is out, then we have class outside."

"If we have class outside, then the sun is out."

Only the first is useful to you when you see the sun come out.

Only the second is useful if you forgot your umbrella.

$p \rightarrow q$

Implication:

p implies *q*

whenever *p* is true *q* must be true

if *p* then *q*

q if *p*

p is sufficient for *q*

p only if *q*

q is necessary for p

p	q	$p \rightarrow q$
T	**T**	**T**
T	**F**	**F**
F	**T**	**T**
F	**F**	**T**

Implications are super useful, so there are LOTS of translations.
You'll learn these in detail in section.

A More Complicated Statement

"Robbie knows the Pythagorean Theorem if he is a mathematician and took geometry, and he is a mathematician or did not take geometry."

Is this a proposition?

We'd like to *understand* what this proposition means.

In particular, is it true?

A Compound Proposition

"Robbie knows the Pythagorean Theorem if he is a mathematician and took geometry, and he is a mathematician or did not take geometry."

We'd like to *understand* what this proposition means.

First find the simplest (atomic) propositions:

p "Robbie knows the Pythagorean Theorem"

q "Robbie is a mathematician"

r "Robbie took geometry"

$(p$ if $(q$ and $r))$ and $(q$ or (not $r))$

$$(p \text{ if } (q \wedge r)) \wedge (q \vee (\neg r))$$

A Compound Proposition

"Robbie knows the Pythagorean Theorem if he is a mathematician and took geometry, and he is a mathematician or did not take geometry."

$$(p \text{ if } (q \wedge r)) \wedge (q \vee (\neg r))$$

p	**"Robbie knows the Pythagorean Theorem"**
q	**"Robbie is a mathematician"**
r	**"Robbie took geometry"**

How did we know where to put the parentheses?

- Subtle English grammar choices (top-level parentheses are independent clauses).
- Context/which parsing will make more sense.
- Conventions

A reading on this is coming soon!

Back to the Compound Proposition…

"Robbie knows the Pythagorean Theorem if he is a mathematician and took geometry, and he is a mathematician or did not take geometry."

$$(p \text{ if } (q \land r)) \land (q \lor (\neg r))$$

p	**"Robbie knows the Pythagorean Theorem"**
q	**"Robbie is a mathematician"**
r	**"Robbie took geometry"**

What promise am I making?

$$((q \land r) \rightarrow p) \land (q \lor (\neg r)) \qquad (p \rightarrow (q \land r)) \land (q \lor (\neg r))$$

The first one! Being a mathematician and taking geometry goes with the "if", knowing the Pythagorean Theorem is the consequence.

Breakout Rooms

We'll use breakout rooms to give you a chance to try problems with other students.

Why? It works!

https://www.pnas.org/content/111/23/8410 a meta-analysis of 225 studies.

Just listening to me isn't as good for you as listening to me then trying problems on your own and with each other.

Breakout Rooms

Every lecture we'll give you an activity to do in the breakout rooms.

Directions are in Activity pdf

Go to cs.uw.edu/311 and get that pdf!

CSE 311: Foundations of Computing I

Announcements

09/30 Welcome to CSE 311! This is our course website — `cs.uw.edu/311`. Most of the materials in this class will be posted here. Visit early and often!

09/29 You should now have access to Zoom links, Gradescope, and Ed. Ping us if anything is missing! Our backup email address is `cse311-staff@cs.washington.edu`.

Calendar

Week 1	Topic	Materials	Assignments
Lecture 1 (Wed 09/30)	Welcome! Propositional Logic	Slides: pptx, pdf Activity: pdf	
Section 1 (Thu 10/01)	Propostions, Translation		
Lecture 2 (Fri 10/02)	Equivalences, Proofs		HW1 out
Week 2			
Lecture 3 (Mon 10/05)	Equivalences, Digital Logic		

Lecture 1 Activity

Introduce yourselves!

If you can turn your video on, please do.

If you can't, please unmute and say hi.

If you can't do either, say "hi" in chat.

Practice filling out a poll everywhere for Activity Credit!

Go to pollev.com/cse-311 and login with your UW identity

Choose someone to share screen, showing this pdf.

Answer these "get to know you" questions until you're pulled back to the main room.

What is your favorite socially-distanced activity?

What class are you most excited about this quarter?

And why is it 311?

Found a new friend? A new study group? Share your emails!

Todo

Tonight:

Pre-course survey on canvas.

Make sure you can access the Ed discussion board

Tomorrow:

Go to section

Soon:

Form a study group!