474 Notes on Day 1 slides:

Slide 7: Why Study the Theory of Computation?

> In this book, we present a theory of what can be computed and what cannot. We also sketch some theoretical frameworks that can inform the design of programs to solve a wide variety of problems. But why do we bother? Why don't we just skip ahead and write the programs that we need? This chapter is a short attempt to answer that question.

What is this document? I have simply copied the instructor notes from the slides that have them, and pasted them in here.

These are some of the things I would say and/or write on the board during class time

Slide 10: Programming in the New Millennium

Today' programmers can't read code from 50 years ago.

Programmers from the early days could never have imagined what a program of today would look like.

Slide 12: More detailed questions

One of the surprising and revolutionary results of the 20th century is there are undecideable problems.

I.e. problems which can be easily stated, but are not computable in finite time for all finite inputs.

We'll explore that in detail later.

Slide 13: Applications of the Theory

There are many more examples in Chapter 1 of the book (and in its many application-oriented appendices)

We will revisit applications once we have some theory to apply. But mainly we will talk about the theory.

Slide 14: What we will focus on

I would guess that 50-60% of homework problems will involve either a proof that something is true or a counterexample to show that some statement is not always true.

Slide 23: Concatenation and reverse of strings (**this was done on the board in class**):

Proof: By induction on $|x|$: This slide is hidden. Do it on the board.

$|x| = 0$: Then $x = \varepsilon$, and $(wx)^R = (w\,\varepsilon)^R = (w)^R = \varepsilon\, w^R = \varepsilon^R\, w^R = x^R\, w^R$.

$$\forall n \geq 0\ (((|x| = n) \rightarrow ((w\,x)^R = x^R\, w^R)) \rightarrow ((|x| = n + 1) \rightarrow ((w\,x)^R = x^R\, w^R))):$$

Consider any string x, where $|x| = n + 1$. Then $x = u\,a$ for some character a and $|u| = n$. So:

$(w\,x)^R$	$= (w\,(u\,a))^R$	rewrite x as ua
	$= ((w\,u)\,a)^R$	associativity of concatenation
	$= a\,(w\,u)^R$	definition of reversal
	$= a\,(u^R\,w^R)$	induction hypothesis
	$= (a\,u^R)\,w^R$	associativity of concatenation
	$= (ua)^R\,w^R$	definition of reversal
	$= x^R\,w^R$	rewrite ua as x

Slide 28: Defining a language: Note that the first two languages are not the same.

Slide 29: Example Language Definitions

The English description is somewhat ambiguous (this is often a problem); the intention is that all of the strings from the first list are actually in the language.

Slide 30: Example Language Definitions:

Answer: The language of all strings that end in *a*

Slide 35: languages and prefixes.

Answers:

$\{a\}^*$

$\{\varepsilon\} \cup \{ bx : x \in \{a, b\}^*:\}$

$\varnothing$