

CSE341: Programming Languages, Spring 2017

[All](#)[Course Info](#)[Contact Info](#)[Course Materials](#)[Homeworks](#)[Exams](#)[Software](#)[Links](#)

Course Information

[Syllabus](#)[Academic-Integrity Policy](#)[Challenge-Problem Policy](#)[Relation to Coursera Course](#)[Gradebook](#)

[Course Calendar](#) (the only information that is on the calendar without also being on the main course page is one-time office-hour changes)

Lecture: Monday, Wednesday, Friday 12:30-1:20 JHN 102

Section AA: Thursday 12:30-1:20, MGH 242

Section AB: Thursday 1:30-2:20, MGH 241

Section AC: Thursday 2:30-3:20, MGH 271

Section AD: Thursday 11:30-12:20, CMU 228

Section AE: Thursday 9:30-10:20, THO 125

Section AF: Thursday 12:30-1:20, CMU 228

Office Hours:

Mondays, 10:30-11:30AM, CSE574, Dan Grossman

Mondays, 4:30-5:30PM, CSE002 (one of the labs), Emily Leland

Tuesdays, 10:30-11:30AM, CSE 007, Justin Harjanto

Tuesdays, 12:00-1:00PM, CSE 021, Nick Mooney

Wednesdays, 9:30-10:30AM, CSE 006 (one of the labs), Waylon Huang

Wednesdays, 1:30-2:30PM, CSE 006 (one of the labs), Miles Saul

Thursdays, 9:30-10:30AM, CSE 220, Spencer Pearson

Thursdays, 3:00-4:00PM, CSE 3rd-floor breakout, Tam Dang

Fridays, 1:30-2:30PM, CSE 021, Daniel Fang

Fridays, 2:30-3:30PM, CSE 021, Thomas Sixuan Lou

Contact Information

[Course Email List](#) (mandatory): You should receive email sent to the course mailing list regularly, roughly at least once a day. Any important announcements will be sent to this list.

Email sent to cse341-staff@cs.washington.edu will reach the instructor and all the TAs. For questions multiple staff members can answer, we encourage you to use this email so that you get a quicker reply and the whole staff is aware of points of confusion.

Course staff:

All staff: cse341-staff@cs.washington.edu

Instructor: Dan Grossman, djg@cs.uw.edu (**not** @u...)

TA: Tam Dang, dangt7@uw.edu

TA: Daniel Fang, danfang@uw.edu

TA: Justin Harjanto, gestone@cs.uw.edu

TA: Waylon Huang, waylonh@cs.uw.edu

TA: Emily Leland, emilyjleland@gmail.com

TA: Thomas Sixuan Lou, lous@cs.uw.edu

TA: Nicholas Mooney, nmooney@cs.uw.edu

TA: Spencer Pearson, suspense@cs.uw.edu

TA: Miles Saul, mhsaul@uw.edu

[Course Discussion Board](#) (optional but encouraged)

Course Materials

Material in the future naturally subject to change in terms of coverage or schedule

Unit 1: ML Functions, Tuples, Lists, and More [Reading Notes](#) [Videos](#)

L1. Mar 27-29: Course Mechanics, ML Variable Bindings slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

L2. Mar 29-31: Functions, Pairs, Lists slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

S1. Apr 1: Emacs, SML Mode, Shadowing, Error Messages

Justin slides: [pptx](#) [pdf](#) [pdf6up](#) Nick slides: [pptx](#) [pdf](#) [pdf6up](#) Spencer slides: [pptx](#) [pdf](#) [pdf6up](#) code: [errors](#) [errors fixed](#)

L3. Apr 3: Local Bindings, Options, Benefits of No Mutation slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

Unit 2: Datatypes, Pattern Matching, Tail Recursion, and More [Reading Notes](#) [Videos](#)

L4. Apr 5: Records, Datatypes, Case Expressions slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

S2. Apr 6: Type Synonyms, Polymorphism, & More

Justin: [pptx](#) [pdf](#) [pdf6up](#) [sml](#) [java](#) Nick: [pptx](#) [pdf](#) [pdf6up](#) [sml](#) Spencer: [pptx](#) [pdf](#) [pdf6up](#)

L5. Apr 7: More Datatypes and Pattern Matching slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

L6. Apr 10-12: Nested Pattern-Matching, Exceptions, Tail Recursion slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

Unit 3: First-Class Functions and Closures [Reading Notes](#) [Videos](#)

L7. Apr 12-14 First-Class Functions slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

S3. Apr 13: Standard-Library Docs, Unnecessary Function Wrapping, Fold & More

Justin: [pptx](#) [pdf](#) [pdf6up](#) [sml](#) [mutual recursion](#) [java](#) Nick: [pptx](#) [pdf](#) [pdf6up](#) [sml](#) Spencer: [pptx](#) [pdf](#) [pdf6up](#) [sml](#)

L8. Apr 14-17: Lexical Scope and Function Closures slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

L9. Apr 17: Function-Closure Idioms slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

Unit 4: ML Modules, Type Inference, Equivalence, & More [Reading Notes](#) [Videos](#)

L10. Apr 19: ML Modules slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

S4. Apr 20: Mutual Recursion, More Currying, More Modules

Justin: [sml](#) [js](#) Nick: [pptx](#) [pdf](#) [pdf6up](#) [sml](#) Spencer: [pptx](#) [pdf](#) [pdf6up](#) [sml](#)

L11. Apr 21-24: Type Inference slides: [pptx](#) [pdf](#) [pdf6up](#) code: [sml](#)

L12. Apr 24: Equivalence slides: [pptx](#) [pdf](#) [pdf6up](#)

Course-Motivation Interlude, Apr 26 [slides](#) [pdf](#) [pdf6up](#) [Videos](#)

S5. Apr 27: Midterm Review

Spencer: [sml](#)

Unit 5: Racket, Delaying Evaluation, Memoization, Macros [Reading Notes](#) [Videos](#)

L13. Apr 26, May 1-3: Racket Introduction slides: [pptx](#) [pdf](#) [pdf6up](#) code: [rkt](#)

L14. May 3-5: Thunks, Laziness, Streams, Memoization slides: [pptx](#) [pdf](#) [pdf6up](#) code: [rkt](#)

Some of the material in L14 will be covered in S6 instead

S6. May 4: More streams, Memoization, and More

Justin: [rkt](#) Nick: [pptx](#) [pdf](#) [pdf6up](#) [rkt](#) Spencer: [rkt](#)

L15. May 5: Macros slides: [pptx](#) [pdf](#) [pdf6up](#) code: [rkt](#)

Unit 6: Structs, Implementing Languages, Static vs. Dynamic Typing [Reading Notes](#) [Videos](#)

L16. May 8: Datatype-Style Programming With Lists or Structs slides: [pptx](#) [pdf](#) [pdf6up](#) code: [rkt](#) [sml](#)

L17. May 10-12: Implementing Languages Including Closures slides: [pptx](#) [pdf](#) [pdf6up](#) code: [rkt](#)

Some of the material in L17 was covered in S7 instead

S7. May 11: Legal ASTs, Macros as Functions, and More

Emily: [pptx](#) [pdf](#) [pdf6up](#) [rkt](#) Justin: [pptx](#) [pdf](#) [pdf6up](#) [rkt](#) Spencer: [rkt](#) Tam: [pptx](#) [pdf](#) [pdf6up](#) [rkt](#)

L18. May 12-15: Static vs. Dynamic Typing slides: [pptx](#) [pdf](#) [pdf6up](#) code: [rkt](#) [sml](#)

Unit 7: Ruby, Object-Oriented Programming, Subclassing [Reading Notes](#) [Videos](#)

L19. May 17: Introduction to Ruby and OOP slides: [pptx](#) [pdf](#) [pdf6up](#) code: [lec19_silly.rb](#) [lec19_example.rb](#)

S8. May 18: Ruby arrays, hashes, ranges, blocks, and more

Justin and Nick: [rb](#)

Spencer: [rb](#)

L20. May 19-22: Arrays & Such, Blocks & Procs, Inheritance & Overriding slides: [pptx](#) [pdf](#) [pdf6up](#) code: [rb](#)

Some of the material in L20 will be covered in S8 instead

L21. May 22: Dynamic Dispatch Precisely, & Manually in Racket slides: [pptx](#) [pdf](#) [pdf6up](#) code: [rb](#) [sml](#) [rkt](#)

Unit 8: Program Decomposition, Mixins, Subtyping, and More [Reading Notes](#) [Videos](#)

L22. May 24: OOP vs. Functional Decomposition; Adding Operators & Variants; Double-Dispatch

slides: [pptx](#) [pdf](#) [pdf6up](#) code stage A: [sml](#) [rb](#) [java](#) code stage B: [sml](#) [rb](#) [java](#) code stage C: [sml](#) [rb](#) [java](#)

S9. May 25: Double-Dispatch, Mixins, and Visitors

Nick: [pptx](#) [pdf](#) [pdf6up](#) [sml dispatch](#) [sml visitor](#) [rb dispatch](#) [rb visitor](#) [rb mixins](#)

Justin: (same code as Nick)

Spencer: (same code as Nick)

L23. May 26: Multiple Inheritance, Mixins, Interfaces, Abstract Methods slides: [pptx](#) [pdf](#) [pdf6up](#) code: [rb](#)

L24. May 31: Subtyping slides: [pptx](#) [pdf](#) [pdf6up](#)

S10. June 1: Final Exam Review

Spencer: [html](#)

L25. June 2: Subtyping for OOP; Comparing/Combining Generics and Subtyping slides: [pptx](#) [pdf](#) [pdf6up](#)

L26. June 2: Course Victory Lap slides: [pptx](#) [pdf](#) [pdf6up](#)

Homework Assignments

[Homework 0](#): on-line survey worth 0 points, "due" Wednesday March 29

[Dropbox for homework turn-in](#)

[Homework 1](#), due Wednesday April 5, 11PM

[Homework 2](#), due Friday April 14, 11PM [hw2.zip](#) (contains 10 files, only 2-3 of which you need to modify)

[Homework 3](#) due Monday April 24, 11PM [provided code](#)

[Homework 4](#) due Tuesday May 9, 11PM [provided code](#) [provided tests](#)

sample image files: [dan.jpg](#) [curry.jpg](#) [dog.jpg](#) [dog2.jpg](#)

[Homework 5](#) due Wednesday May 17, 11PM [provided code](#) [provided tests](#)

[Homework 6](#) due Thursday May 25, 11PM

[hw6graphics.rb](#) [hw6provided.rb](#) [hw6runner.rb](#) [hw6assignment.rb](#)

[Homework 7](#) due Friday June 2, 11PM

[hw7.sml](#) [hw7.rb](#) [hw7testspublished.sml](#) [hw7testspublished.rb](#)

Exams

Midterm: Friday April 28, in class [unsolved](#) [solved](#)

Sample midterms:

Spring 2016 [unsolved](#) [solved](#)

Spring 2013 [unsolved](#) [solved](#)

Winter 2013 [unsolved](#) [solved](#)

Fall 2011 [unsolved](#) [solved](#)

Spring 2011 [unsolved](#) [solved](#)

Spring 2008 [unsolved](#) [solved](#)

Winter 2008 [unsolved](#) [solved](#)

Final Exam: Thursday June 8, 8:30-10:20AM [unsolved](#) [solved](#)

Sample finals:

Spring 2016 [unsolved](#) [solved](#)

Spring 2013 [unsolved](#) [solved](#)

Winter 2013 [unsolved](#) [solved](#)

Fall 2011 [unsolved](#) [solved](#)

Spring 2011 [unsolved](#) [solved](#)

Spring 2008 [unsolved](#) [solved](#)

Winter 2008 [unsolved](#) [solved](#)

Software Installation and Use

[Instructions for SML and Emacs](#), which is everything you need for the first half of the course.

[Videos](#) showing the software installation on Windows

[Instructions for Racket and DrRacket](#)

[Instructions for Ruby and irb](#)

Links to Other Resources

The course materials on this page (lectures, sections, homeworks, installation instructions, videos) are designed to provide what you need for the course except for some details that you can look up in standard-library documentation or users' guides for particular languages. Links for such information is below. We also provide links to useful books and tutorials that provide alternate explanations. We will not follow any textbooks closely, but you may still find them useful. Suggestions for additional links are welcome.

SML resources:

www.smlnj.org (links to many things, including the next three resources)

[user's guide](#)

[standard-library documentation](#)

[tutorials, books, and documentation](#)

[Elements of ML Programming, ML'97 Edition](#), Jeffrey D. Ullman, 1998.

This is a textbook that takes a different approach but does cover some of the same material.

Check the [errata page](#) to avoid bugs.

Racket resources:

[The Racket Guide](#)

racket-lang.org, particularly the Docs button

Ruby resources:

[Programming Ruby 1.9 & 2.0: The Pragmatic Programmers' Guide](#), Dave Thomas et al.

Check the [errata page](#) to avoid bugs.

ruby-doc.org

[Ruby home page](#)

 [Valid CSS!](#)  [Valid XHTML 1.1](#)