

Lecture 1: Course Overview

CS 454: Principles of Programming Languages

Spring 2025

Acknowledgments

Course materials originally developed by [Prof. Sriram Sankaranarayanan](#) for [Principles of Programming Languages \(CSCI 3155\)](#) at University of Colorado Boulder

Also referenced course materials developed by:

- [Prof. Bor-Yuh Evan Chang](#) at University of Colorado Boulder
- [Prof. Shriram Krishnamurthi](#) at Brown University
- [Prof. Steven Holtzen](#) at Northeastern University
- [Prof. Adrian Sampson](#) at Cornell University
- [Prof. Lindsey Kuper](#) at UC Santa Cruz

Agenda

- What is the course about?
- Course structure
- Course logistics
- Course policies

What is the course about?

What this course is NOT about?

- A survey of different programming languages

- An introduction to programming

What is the course about?

A deep-dive into the questions:

- What is a programming language?
- How to implement a programming language?

Along the way, you will learn:

- How to use a functional programming language (specifically, Scala)?
- How to write programs that manipulate programs?

What is a programming language?

(Wikipedia) A **programming language** is a system of notation for writing computer programs

A programming language is a means of unambiguously conveying a sequence of instructions to a computer

Programming languages have two parts

Syntax (Form)

```
x=5  
print(x)
```

What does it look like?

Semantics (Meaning)

- Create a local variable called “x” and assign it equal to 5
- Print the value of “x” to the console

What does it do?

This course is all about precisely defining programming languages!

Syntax (Form)

Formal descriptions
as grammars

What does it look like?

Semantics (Meaning)

Programs that run
programs, i.e.,
interpreters!

What does it do?

In this course...

- We will study a series of increasingly expressive toy programming languages
- We will build an interpreter for each of these programming languages
- All interpreters will be implemented in Scala, a functional programming language

Why study programming languages (PLs)?

- Learn about the fundamental building blocks and features of PLs
 - Become equipped to learn new languages quickly
- Learn how to design and implement PLs, if needed
 - Buried in every large, extensible software system, is a small domain-specific language (DSL), for instance, Macros in Excel
- Become a more effective programmer
 - Learn how to choose languages for your problem
 - Learn how to manipulate programs as data
 - Learn how to think about programs in a rigorous way

Course Structure

Course structure

- Introduction to Scala, a functional programming language
- Build an interpreter in Scala for a small toy programming language, Lettuce
- while (Not end of semester):
 - Add a new language feature to Lettuce and build an interpreter for this version of Lettuce

Course Logistics

Course Staff

- Instructor: Ravi Mangal
 - Assistant Professor at CSU since Fall 2024
 - Research in analyzing safety and correctness of software systems including AI-based systems
 - ravi.mangal@colostate.edu
- TA: Jonathon Yallop
 - Masters in CS student at CSU
 - Research in programming languages, compiling parallel programs
 - Jonathon.Yallop@colostate.edu

Course logistics: Communication

- Course website - <https://www.cs.colostate.edu/~cs454/>
 - Includes syllabus, schedule, and course policies
 - Please read carefully!
 - Slides and course notes will also be uploaded here
- [Canvas](#) for announcements, assignment submissions, and course materials including slides and course notes
- [Teams](#) for discussion

Course logistics: Communication

- Office hours:
 - Ravi: Wednesday (4-5 PM) and by appointment
 - Jonathon: Tuesday (4-6 PM), Wednesday (10 AM-12 Noon), Thursday (10 AM-12 Noon)
 - Let us know if none of these office hours work for you
- Email:
 - ravi.mangal@colostate.edu
 - Jonathon.Yallop@colostate.edu
 - Include CS454 in subject!

Anonymous feedback here

Course logistics: Grading break-up

Weekly Assignments (About 12)	50%
Quizzes	20%
Final	25%
Class Participation and Attendance	5%

Course logistics: Grading

The quizzes will be conducted in the classroom during regular lecture hours. You will be informed about an upcoming quiz a few days in advance. To get full grades on class participation and attendance, you have to do the following:

- Introduce yourself on Teams in the first week of the course.
- Fill out a mid-semester feedback survey.
- Be present in the class for **5** roll-calls that will be conducted randomly during the semester. In the event that unplanned absences occur due to illness, family emergency, etc., you can seek an attendance allowance from the instructor. It is best to inform the instructor in advance, although we understand that this is not always possible.

Active engagement in the classroom and on Teams is highly encouraged!

Course logistics: Grading thresholds

Score Range	Letter Grade
≥ 97	A+
≥ 93	A
≥ 90	A-
≥ 87	B+
≥ 83	B
≥ 80	B-
≥ 77	C+
≥ 73	C
≥ 70	C-
≥ 67	D+
≥ 63	D
≥ 60	D-
< 60	F

- Grades maybe curved up based on course circumstances.

Course logistics: Workload

- 1 assignment, roughly every week
 - Due on Thursday nights at 11:59 PM
- 1 in-class quiz, roughly every two-three weeks
- In-person final exam

Course logistics: Software to be used

- Scala and Jupyter Notebooks
- Everything will be setup inside a Docker container
- Please follow instructions [here](#) to setup your local machine

Course Policies

Late Work Policy

Assignments will be due at 11:59 PM on the dates specified in Canvas and the course [schedule](#). We ask that you try your best to submit assignments on time, even if your solution is incomplete. Any late work will be penalized as follows:

- Less than 24 hours late: 5% penalty on the maximum possible grade for the assignment
- More than 24 hours late: Submission not accepted

Late submissions without penalty and extensions beyond 24 hours will only be granted on a case-by-case basis in exceptional circumstances. Such extensions are more likely when permission is sought in advance, for reasons which are unexpected and beyond your control, and which involve only a short extension. For any such extension, appropriate documentary evidence will be required. Even under such exceptional circumstances, the instructor reserves the right to assign a score penalty to late work, depending on the circumstances and degree of lateness.

Collaboration Policy

All the solutions you turn in for the assignments must be completely your own work. However, you are welcome to work together in learning the material covered in the lectures. You are also allowed to discuss assignments with other students. But the final answers and code you submit must explicitly be your own work. Specifically,

- Do not show any partial solution to another student.
- Never share code with other students or on a public-facing website. (Shared code is surprisingly easy to detect.)
- Do not search the Internet for solutions. Don't search on Stack Overflow or anywhere else.
- Do ask someone if you're confused about what the assignment is asking for.
- Cite any resources such as web sites, academic papers, or books used to develop your solution. Similarly, acknowledge anyone with whom you discussed the assignment.
- Definitely ask the course staff if you're not sure whether or not something is OK.

Note that these guidelines do not prevent you from discussions with other students. For example, you may discuss the assignment at the whiteboard with another student, but then you must erase the whiteboard, go home, and write up your solution individually (while also acknowledging the person with whom you discussed).

Generative AI Policy

Generative AI tools such as ChatGPT or Claude are increasingly powerful and might be capable of solving some of the assignments. While you may use these tools to understand the course material, the assignments must be your own work and the use of these tools for generating assignment solutions is strictly prohibited (will be treated as a violation of academic integrity). There are multiple reasons why these tools are detrimental to your learning experience:

- They rob you of the ability to think and learn the concepts for yourself since solving problems is an essential step to gaining a solid understanding of the material.
- You will struggle with the in-classroom quizzes and exams where you will not have access to these tools.
- While we acknowledge that these tools are likely to become an important part of a software engineer's workflow in the future, you are much more likely to use these tools in an effective manner if you already have expertise in the relevant technical topics. Developing such expertise requires putting in the effort to learn these topics without the assistance of these tools.
- These tools are prone to generating imperfect or even incorrect solutions, so trusting them blindly can lead to bad consequences.

Ultimately, you will get out of the class what you put in. Simply copying and pasting code from ChatGPT/Claude will not lead to any learning. If you are stuck on the assignments, reach out to the course staff or your fellow students. The course staff is eager to help you!

Disability and Wellness

- Accommodations provided for document disabilities
 - Get verifying memo from the [Student Disability Center](#) (SDC) in a timely manner
- If you are experiencing undue personal or academic stress at any time during the semester (or if you notice that a fellow student is), contact me, your academic advisor, or CSU Mental Health services.

Due this week

- Post a note on Teams introducing yourself and your background
 - What's your background?
 - Any experience with functional programming?
 - Experience with other programming languages. Which languages? How many years?
 - What do you want out of this class?
 - If you prefer, you can send a separate private note to me about your background.
- Go through the details on the course [website](#)
- Setup required software on your machines. Follow instructions [here](#)