Syllabus - Principles of Programming Languages - CSCI 3155 - Fall 2024

Course Overview and Goals

This course is about principles, concepts, and ideas that underlie programming languages. But what does this statement mean?

As a student of computer science, it is completely reasonable to think and ask, “Why bother? I’m proficient and like programming in Ruby. Isn’t that enough? Isn’t language choice just a matter of taste? If not, should I be using another language?”

Certainly, there are social factors and an aspect of personal preference that affect the programming languages that we use. But there is also a body of principles and mathematical theories that allow us to discuss and think about languages in a rigorous manner. We study these underpinnings because a language affects the way one approaches problems working in that language and affects the way one implements that language. At the end of this course, we hope that you will have grown in the following ways:

- You will be able to learn new languages quickly and select a suitable one for your task.
- You will gain new ways of viewing computation and approaching algorithmic problems.
- You will gain new ways of viewing programs.
- You will gain insight into avoiding mistakes for when you design languages.
- You will be able to use AI assistants to accelerate your creative design.

We will *dissect* programming languages by constructing interpreters. The lab assignments are to construct interpreters for mini-versions of JavaScript. We will see that interpreters are the basis for realizing computation, and we will study the programming language theory that enable us to reason carefully about a language’s design and implementation. Our approach will be gradual in that we will initially consider a small subset of JavaScript and then slowly grow the aspects of the language that we consider (and see how they underlie many other programming languages).

Incoming students often expect this course to be what I will call a trip to the Zoo of Programming Languages. While it is certainly interesting to go to the zoo, we seek a more informative and scientific study of the underlying principles. A more apt analogy is an **anatomy** course where we will study the "guts" and inner-workings of programming languages. After this course, such an anatomical study will enable us to compare and contrast programming languages in a substantive manner and address the learning goals outlined above.

The course covers many aspects of using, understanding, and reasoning about programming languages (e.g., syntax, scoping, induction, data types, and typing). We will build up a set of mathematical tools for careful discourse. A significant part is devoted to abstraction, that is, how languages help programming in the large (e.g., subtyping polymorphism, parametric polymorphism, modules, and objects).

This course prepares you for introductory courses on both programming language implementation (e.g., CSCI 4555) and programming language semantics (e.g., CSCI 5535).

Prerequisites

The prerequisites for this course are CSCI 2270 (Computer Science 2: Data Structures), CSCI 2824 (Discrete Mathematics), and CSCI 2400 (Computer Systems).

If you have not already taken these courses or if you have any concerns, please talk with the Instructor. Proficiency in programming is absolutely expected, though there is no specific language required. This means that you should be able to pick up a new programming language with relative ease. You will be expected to learn new programming languages with guidance but relatively independently in this course.

Requirements

The course consists of the following learning activities:

- **Reading**. There will be required articles or book chapters to read. This course will emphasize *learning from the resources out of class and actively discussing problems and questions in class.*
- **Quizzes and Exercises** (25%). There will be online quizzes and homework exercises associated with each unit. *There will be no make-ups if you miss a quiz/exercise, but your lowest quiz/exercise grade will be dropped.*
- **Lab Assignments** (30%). This course is an active-learning course, so most of the understanding will come from attacking the lab assignments. Lab assignments are the capstone of each unit, which are generally two weeks in length. Because iteration on lab assignments is the basis for learning in this course, we offer a generous "redo" policy on lab assignments (see below). *No late assignments will be accepted, but your lowest lab grade will be dropped.*
- **Class Participation** (5%). Participation includes both in-class and online discussion. See below for expectations for class participation.

And the following summative assessments:

- **In-Class Exams** (20%). We will have at least five "spot" exams conducted at the end of the class. Each exam will be announced 2-3 days before it is conducted. These exams will consist of the material covered in class or through your assignments. *All students have a built-in accommodation who have to miss an exam due to unforeseen circumstances, as your lowest in-class exam grade will be dropped. There will be no make-ups.*
- **Final Exam** (20%). The final exam will be at a time announced by the University registrar.

Grading

Your overall grade will be determined using the ratio for lab assignments, exercises, class participation, the midterm exam, and the final exam shown above.

Regrades

Any concern about an error in grading must be submitted within **one week** of when it is returned. Any coursework submitted for reconsideration may be regraded in its entirety, which could result in a lower score if warranted. To request a regrade, please go to the Course Manager's office hours with your coursework and an explanation of what you believe the grading error to be.

Make-Up Policy

There will be no special or make-up examinations or assignments for any student (except in the case of emergency or the stated religious observances).

Redo Policy

This course is an active learning course, which means the learning is driven primarily by active discovery in doing the assignments. To encourage iteration until mastery, you may "redo" any assignment via an oral interview with an instructor for a maximum of 90%. In a "redo interview," the instructor is probing for your level of understanding and may ask different questions than you have seen previously. A "redo" must be completed within **one week** of when the assignment is returned. You may request one interview per assignment. However, you may discuss your solutions with the instructors in office hours as much as you like before requesting your regrade. You must submit your assignment on time to participate in a "redo". To request a "redo" interview, please contact the Course Manager to schedule a time with a "redo" interviewer from the Instructional Staff (i.e., either an Instructor or a TA).

Extra Credit and Participation

Extra credit opportunities may be offered during the course semester. Extra credit is recorded separately from normal grades and are only considered after final grades have been calculated. If your final grade is just below a grade cutoff, extra credit may bump you up to the next grade. Finding a bug in the course materials that is then adopted is a standing offer for extra credit.

Teams and Pair Programming

Your goal as a team is to help each other successfully complete the course assignments and the course together. You may be asked to work on assignments in *pairs*, enabling pair programming. You are also asked to work with a different partner for each unit, enabling the benefit from the experience of your team. You will get your partner in your lab section. Assignments are the main opportunity to learn material in this course and thus they are graded relatively generously. It is strongly advised that you work on all the problems in an assignment together so that you understand all of the material and are prepared for the exam. Everyone will submit assignments, and you must cite your partner explicitly. You are responsible for all assignments individually (e.g., if your partner drops the course midway though an assignment, you still need to submit on time).

Autotesting

We will use automated testing as way to provide you with quick, iterative feedback on your progress in achieving the learning outcomes. Automated testing will be used minimally for evaluation.

Workload

CSCI 3155 is a 4-credit course. **This translates to the University expectation of approximately 8 hours of outside of class time per week or 16 hours per two-week unit.**

Evaluation

Both your ideas and also the clarity with which they are expressed matter—both in your English prose and your code!

We will consider the following criteria in our grading:

- *How well does your submission answer the questions*? For example, a common mistake is to give an example when a question asks for an explanation. An example may be useful in your explanation, but it should not take the place of the explanation.
- *How clear is your submission*? If we cannot understand what you are trying to say, then we cannot give you points for it. Try reading your answer aloud to yourself or a friend; this technique is often a great way to identify holes in your reasoning. For code, not every program that "works" deserves full credit. We must be able to read and understand your intent. Make sure you state any preconditions or invariants for your functions (either in comments, as assertions, or as require clauses as appropriate).

Expectations for Class Participation

Classes focus on discussion, so it is certainly important to participate in class and read the readings beforehand.

Students are expected to be active participants in class and in the online forums (e.g., around at least 1 substantive post per week). You may and are encouraged to post comments or questions about the reading before the class where we will cover it. Posting early will help focus our discussion.

Here are some examples of good comments:

- Questions about the reading or the class discussion.
- Thoughtful answers to other people's questions.
- Clarification of some point discussed in class.
- Thoughtful comments on a web resource related to a reading or class discussion that caught your attention (e.g., you happened upon an interesting blog post related to our discussion). Note that simply giving a link is not considered a thoughtful comment.

If everything from a class meeting seems clear to you, try to come up with a question that tests your classmates' understanding. Put yourself in the position of an Instructor!

Overall, the intent is for you to to take a moment to reflect upon the day's reading or class discussion. The course staff will read all posts but may not respond to all of them (e.g., if we believe your classmates' answers are sufficient).

Textbook and Resources

Textbook

The primary reading for the course is the course notes: Principles and Practice in Programming Languages.

The course follows some ideas from a supplemental text: Essentials of Programming Languages, 3rd edition by Daniel P. Friedman and Mitchell Wand. We will take liberty to deviate this text where appropriate, as we will use a different implementation language (Scala instead of Scheme). This book is available as an e-book to all University students.

Prof. Sankaranarayanan has authored a similar set of course notes following the design of this course.

We also strongly recommend that you get access to Programming in Scala, 4th edition by Martin Odersky, Lex Spoon, and Bill Venners. This book is an extended tutorial for learning Scala by those directly involved in the language's development. This book is available as an e-book to all University students.

Two other useful Scala texts: Atomic Scala by Bruce Eckel and Dianne Marsh; and Functional Programming in Scala by Paul Chiusano and Runar Bjarnason (available as an e-book to all University students).

Scala

- Language Website. The central point for Scala information. The reference implementation for this course is Scala 2.13.x.
- Metals for VS Code. You are welcome to use any code development environment, but the course staff will only support Metals for VS Code.ntelliJ. There are many options, including IntelliJ IDEA, Emacs, Eclipse, and NetBeans.
- Scala Style Guide. We will in general follow the Scala style guide.

Git

There are many guides and tutorials on using git and GitHub. Here are some links to start with:

- Hello World from GitHub Guides.
- Syncing from Atlassian Tutorials.
- Pro Git by Scott Chacon and Ben Straub. This free e-book gives a good basis for understanding git.

JavaScript

- node.js. A JavaScript platform for server applications based on Google Chrome's V8 JavaScript engine. We recommend installing node.js to try out small JavaScript examples on the command-line. Node.js v18.7.0 will serve as our reference JavaScript implementation.

- JavaScript: The Good Parts by Douglas Crockford.
- Mozilla JavaScript overview and reference.
- ECMA-262 standard specification

Tools

Website

The primary channel for course materials is the course website that will link to other resources.

Canvas

We use Canvas primarily for managing grades. You should have been automatically enrolled. If you know of someone that does not have the necessary Canvas access, please direct them to the Course Manager to resolve the matter.

Gradescope

We use Gradescope for assignment submissions. You enroll in Gradescope through Canvas.

GitHub

We use GitHub for distribution of some programming assignments. You create your GitHub assignment repositories through Canvas.

Piazza

We use Piazza for online, outside-of-class discussion. Rather than emailing questions to the course staff, questions should be posted on Piazza. I encourage you to make class-wide posts whenever possible, but there is an option to send an instructor-private message. You enroll with Piazza through Canvas.

Computing

You may use the Department's cloud-computing environment coding.csel.io. We strongly suggest you use a Unix-based development environment as a back up (e.g., Ubuntu Linux, macOS). While not officially supported, we will try to help with other Scala installs. Please feel free to ask for help on the forum and help by contributing your experience.

Integrity of the Course Materials

The development effort in the course materials, including the lab assignments, the exercises, and the exams, is significant. You agree that you will not share any course materials publicly or privately outside of your teams. The course materials, include your or anyone else's solutions to the lab assignments, exercises, and exams. In particular, you agree not to post your solutions to the lab assignments in a public source code repository, such as public GitHub repositories. Use the private source code repositories provided to you for your work.

Note that there is no conflict with the Collaboration Policy described below. You are welcome and encouraged to support each other in the learning of the material.

Collaboration Policy

You are welcome and encouraged to work together in learning the material. If you worked with someone on an assignment, or if your submission includes quotes from a book, a paper, or a web site, you should thank the source. Bottom line, feel free to use resources that are available to you as long as the use is **reasonable** and you **cite** them in your submission. However, **copying answers directly or indirectly from solution manuals, web pages, your peers, or even a course staff member is certainly unreasonable**. If you have any doubts in this regard, please ask the course staff.

You are allowed to discuss homework assignments with other students. However, in order to ensure that the work you submit is still your own, we insist that you adhere to a *whiteboard policy* regarding these discussions: you are not allowed to take any notes, files, or other records away from the discussion. For example, you may work on the homework at the whiteboard with another student, but then you must erase the whiteboard, go home, and write up your solution individually. We take your ability to recreate the solution independently as proof that you understand the work that you submit.

This policy is our attempt to balance the tension between the benefits of group work and the benefits of individual work. We ask that you obey the spirit of the policy, as well as the letter: ensure that all work you submit is your own and that you fully understand the solution. This is in your best interest: the exam constitutes a significant part of your final grade and it will draw heavily on the terminology, concepts, and techniques that are exercised in the homework. It is unlikely that you will be able to do well on the exam if you do not take full advantage of the learning opportunity afforded by the homework assignments.

Academic Dishonesty Policy

We will go by the Honor Code set forth by the University. Academic sanctions may include, but is not limited to, a zero for the assignment or a failing grade for the course.

Generative AI

We note that tools such as ChatGPT or Github Copilot can solve some of the assignments, though they are going to be imperfect solutions. For your benefit, we discourage you from using them: (a) they will rob you of the ability to think through solutions to simple problems and thus your ability to learn the principles that will enable you to leverage them effectively; (b) they will not be available to you during examinations and you will definitely be at a disadvantage there. You should wait to gain some expertise in Scala and the principles we are teaching in this class. These tools are not going to go away and ultimately learning the principles from this course will make you more productive with them as a professional engineer. Like all tools (e.g., a calculator for arithmetic), using them effectively requires you to develop the expertise and creativity needed without relying on their help.

Discord Channels

Discord and other non-University channels are fine to have, but please include at least one TA in each such channel so that they can vouch that no cheating occurred through these channels.

In summary,

1. *Cite your sources.*
2. *Inspiration is free.* You may discuss homework assignments with anyone. You are especially encouraged to discuss solutions with your instructor and your classmates.
3. *Plagiarism is forbidden.* The assignments and code that you turn in should be written entirely on your own. Copying/soliciting a solution to a problem from the internet or another classmate constitutes a violation of the course's collaboration policy and the Honor Code and will result in an F in the course and a trip to the Honor Council.
4. *Do not search for a solution online or via AI.* You may not actively search for a solution to the problem from the internet. This includes posting to sources like StackExchange, Reddit, Chegg, etc. Searching for basic techniques in programming is totally fine. If you want to post and ask, "How do I group by two columns, then do something, then group by a third column?", that's fine. What you cannot do is post, "Here's the problem my prof gave me. I need to convert Age in Earth years to Martian years and then predict the person's favorite color. Give me code!" That's cheating.
5. *When in doubt, ask.* We have tried to lay down some rules and the spirit of the collaboration policy above. However, we cannot be comprehensive. If you have doubts about this policy or would like to discuss specific cases, please ask an instructor. If it has not been described above, you should discuss it with us first.

Classroom Norms

We trust and expect everyone to behave in a civil and courteous manner.

In class, the course staff promises their undivided attention and reciprocally expects the same from you. In today's world, this promise requires **turning off transmitting devices**, such as mobile phones and wi-fi on notebook computers. The use of notebook computers should be discussed with the instructor and they should be used only for purposes directly relevant to the class discussion. Please notify the course staff if you encounter behavior that distracts from your learning.

Classroom Behavior

We will also go by the policies set forth by the University:

> Both students and faculty are responsible for maintaining an appropriate learning environment in all instructional settings, whether in person, remote or online. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy.
>
> For more information, see the classroom behavior policy, the Student Code of Conduct, and the Office of Institutional Equity and Compliance.

Accommodation for Disabilities, Temporary Medical Conditions, and Medical Isolation

We will also go by the policies set forth by the University:

> If you qualify for accommodations because of a disability, please submit to your professor a letter from Disability Services in a timely manner so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities. Information on requesting accommodations is located on the Disability Services website. Contact Disability Services at 303-492-8671 or dsinfo@colorado.edu for further assistance. If you have a temporary medical condition, see Temporary Medical Conditions on the Disability Services website.
>
> If you have a temporary illness, injury or required medical isolation for which you require adjustment, please contact the course staff to discuss your situation.

Preferred Student Names and Pronouns

We will also go by the policies set forth by the University:

> CU Boulder recognizes that students' legal information doesn't always align with how they identify. Students may update their preferred names and pronouns via the student portal; those preferred names and pronouns are listed on instructors' class rosters. In the absence of such updates, the name that appears on the class roster is the student's legal name.

Honor Code

We will also go by the policies set forth by the University:

> All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the Honor Code. Violations of the Honor Code may include but are not limited to: plagiarism (including use of paper writing services or technology [such as essay bots]), cheating, fabrication, lying, bribery, threat, unauthorized access to academic materials, clicker fraud, submitting the same or similar work in more than one course without permission from all course instructors involved, and aiding academic dishonesty. Understanding the course's syllabus is a vital part in adhering to the Honor Code.
>
> All incidents of academic misconduct will be reported to Student Conduct & Conflict Resolution: StudentConduct@colorado.edu. Students found responsible for violating the Honor Code will be assigned resolution outcomes from the Student Conduct & Conflict Resolution as well as be subject to academic sanctions from the faculty member. Visit Honor Code for more information on the academic integrity policy.

Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

We will go by the policies set forth by the University:

> CU Boulder is committed to fostering an inclusive and welcoming learning, working, and living environment. University policy prohibits protected-class discrimination and harassment, sexual misconduct (harassment, exploitation, and assault), intimate partner abuse (dating or domestic violence), stalking, and related retaliation by or against members of our community on- and off-campus. The Office of Institutional Equity and Compliance (OIEC) addresses these policies, and individuals who believe they have been subjected to misconduct can contact OIEC at 303-492-2127 or email CUreport@colorado.edu. Information about university policies, reporting options, and support resources can be found on the OIEC website.

> Please know that faculty and graduate instructors must inform OIEC when they are made aware of incidents related to these policies regardless of when or where something occurred. This is to ensure that individuals impacted receive outreach from OIEC about resolution options and support resources. To learn more about reporting and support for a variety of concerns, visit the Don't Ignore It.

Religious Observances

We will go by the policies set forth by the University:

> Campus policy requires faculty to provide reasonable accommodations for students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. Please communicate the need for a religious accommodation in a timely manner.
>
> See the campus policy regarding religious observances for full details.

We will try to accommodate religious conflicts in a reasonable manner. Please check the exam dates and submit all requests for adjustments within the **first four weeks** of class. To discuss a special accommodation, please contact the Course Manager with your request.

Mental Health and Wellness

> The University of Colorado Boulder is committed to the well-being of all students. If you are struggling with personal stressors, mental health or substance use concerns that are impacting academic or daily life, please contact Counseling and Psychiatric Services (CAPS) located in C4C or call (303) 492-2277, 24/7.
>
> Free and unlimited telehealth is also available through Academic Live Care. The Academic Live Care site also provides information about additional wellness services on campus that are available to students.

CU Community of Care

> CU Boulder is committed to a community of care in which students are supported by faculty and staff throughout their college journey. You don't have to face academic challenges alone – CU and the college are here to help you learn and succeed in your coursework and campus life. Part of this community of care is your connection to faculty and staff across campus. Our college promotes and hopes you will connect with faculty or staff who may reach out during your educational journey at CU.

Course Alerts

> This course participates in the CU Course Alert process to help connect you with support resources and identify your barriers to success

If you receive a course alert for this class, please see the course staff in office hours or schedule a meeting.