COLLEGE OF ENGINEERING, UIC

CS 476: Programming Language Design (CRN: 32644), UG, 3 Credits

CS 476: Programming Language Design (CRN: 32645), Graduate, 4 Credits

I. Instructor & Course Details

Saeid Tizpaz-Niari
Email address: saeid@uic.edu

Drop-In Office Hours: 1-2 pm Tuesday
Drop-In Hours location:
https://uic.zoom.us/j/88275345971?pwd=BlP0RtziEFkVMRaBVkGCjoeeDHVFbt.1
Meeting ID: 882 7534 5971
Passcode: 6fzVL0Dh

Abhishek Soni
Email address: asoni24@uic.edu

Drop-In Office Hours (in-person or virtual): TBA
Drop-In Hours location: TBA

Blackboard Course Site

https://uic.blackboard.com/webapps/blackboard/content/listContentEditable.jsp?content_id=_14093757_1&course_id=_302824_1&mode=reset#
Students are expected to log into the course site regularly to learn about any developments related to the course, upload assignments, and communicate with classmates. For all technical questions about Blackboard, email the Learning Technology Solutions team at LTS@uic.edu.

Course Modality and Schedule

Class meets on Monday/Wednesday from 3:30 pm to 4:45 pm at CDRLC#1406.

II. Course Information

The purpose of this course is to learn how to think about, describe, and reason about the features of different kinds of programming languages. We will cover the theoretical frameworks that are used to define and describe the behavior of different languages, and build interpreters that implement pieces of those languages. We will cover the key features of imperative, functional, and logic languages, as well as concurrency, I/O, and various other features.

In previous courses, students have examined how to write programs in individual languages such as C, Java, or Python. In this class, we will take a broader view of programming languages, and study the key concepts and techniques that allow developers to implement languages such as Java or C.

Certainly, there are social factors and personal preferences to choose and use programming languages. 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.
We will dissect programming languages by constructing interpreters. The semester project is to construct an interpreter for a toy language called Lettuce (a language based on a family of programming languages called ML). 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. 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.

PREREQUISITES:
The main prerequisite for this class is CS 341. It will also help to be familiar with propositional logic, induction, and formal proofs, as taught in CS 201, MCS 361, or equivalent.

Growth Mindset:
Course materials and assignments can be complex and challenging, but they are crucial to your intellectual and personal growth and development. There are times you may need extra help. Students who attend class consistently, complete all assignments, thoughtfully engage with feedback on work, develop good study strategies, visit the tutoring center, and contact faculty when they are struggling can develop a thorough understanding of the course material and ultimately succeed in the course!

Course Goals and Learning Outcomes

Upon the completion of course:

- formally describe the syntax and semantics of programming languages
- build interpreters for new languages and extend existing languages with new features
- use a range of mathematical tools for specifying program behavior
- understand the features of different language paradigms, including imperative, functional, object-oriented, and logic
- formally describe type systems and implement type checking and inference.

Required and Recommended Course Materials

The primary reading for the course is the Jupyter notebooks that will be distributed each week. Students are required to install Jupyter notebook with Scala kernel (First, install Jupyter and then install Almond that provides Scala kernel supports for Jupyter and available for Mac, Linux, and Windows). Please

find more information on installation in the ``install`` folder in Blackboard. The course follows many ideas from supplemental texts such as Robert W. Sebesta, *Concepts of Programming Languages*, 12th edition, Addison-Wesley, 2018 and *Essentials of Programming Languages (Third Edition)* by Friedman & Wand. **These books are recommended but are not required.** We also strongly recommend (but not required) that you get access to Programming in Scala, third 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.

Required Technology

Ensure your UIC e-mail account is working and that you have access to the Web and a stable web browser. Mozilla Firefox and Google Chrome are the most supported browsers for Blackboard; other browsers may cause complications with the LMS. If you encounter technical difficulties beyond your scope of troubleshooting, please contact the Help Desk as they are trained specifically in assisting with technological needs of students.

Respect for Copyright

Please protect the copyright integrity of all course materials and content. Please do not upload course materials not created by you onto third-party websites or share content with anyone not enrolled in our course.

III. COURSE POLICIES & CLASSROOM EXPECTATIONS

This class will have pre-recorded lectures (~ 10 mins) sent to students at the beginning of week. Students are required to go through the videos and corresponding readings. The class time will devote for discussions, clarifying issues, answering questions, and practicing problem sets. Please bring your laptop to the class to participate in different activities and write code! The course work is divided into four parts: (a) weekly assignments (online quiz and problem set); (b) regularly scheduled spot exams (c) mini projects that consist of a programming assignment using principles learned in this class; and (d) the final exam or final project.

Category	Percentage
Weekly Assignments	40%
Online Quiz	10%
Mini Projects	15%

Spot Exams	20%
Final Project or Exam	15%
Class/group activity	5% (extra points)

Weekly Online Quiz (10% of the grade)

You will have two attempts for each quiz. You can also check your answers for each question, but a wrong answer will involve a small penalty to your grade. ***The lowest grade for the weekly online quiz will be dropped. Therefore, no late excuses will be granted.***

Weekly Problem Sets (40% of the grade)

Each week you will receive a problem set that will include concept-based problems and programming assignments in Scala. We project that there will be **8 weekly assignment**. These assignments will be posted as jupyter notebooks that need to be filled out by the students. The students have to present their solutions to the TA. ***The lowest grade for the assignments will be dropped.***

Mini Projects (15% of the grade)

We will have **two mini projects** that will involve building interesting applications using the skills you have learned in class. One of the mini-projects will implement a language from scratch. Another mini project will explore building a domain specific language. ***No late submissions will be accepted (pay attention to the deadlines for mini-projects).***

Spot Exams (20 % of the grade)

We will have **three spot exams** that will be conducted during your recitations. These exams will last roughly 40 minutes and consist of material covered in class or through your assignments***.***

Final Project or Final Exam (15 % of the grade)

CS 476 will have a final project in place of a final exam for graduate students. The goal of the project is to use the techniques of the class to describe a new programming language or language feature. Students can work alone or in groups of at most 3 students. Undergraduate students can choose to do the final project or take a final exam. More information about the final project will be provided.

Extra Bonus (5 % of the grade)

A bonus of up to 5% is available for engaging and participating in class discussions. Especially, students who are not able to participate in a class discussion are encouraged to post their questions and participate in the class discussions offline. A full bonus requires exceptional participation.

Your grade is independent of anyone else's grade in this class; that is, we do not grade on a curve. Everyone can get an *A* in this class. Our purpose in grading is to uphold a standard of quality and to give you feedback, it is not to rank students.

Percentage	Grade
P >= 90%	A
80% <= P < 90%	B
70% <= P < 80%	C
60% <= P < 70%	D
P < 60%	F

The instructor reserves the right to adjust these criteria downward, e.g., so that 88% or higher represents an A, based on overall class performance. The criteria will not be adjusted the other way around.

Policy for Missed or Late Work

No late submission will be accepted, except for weekly assignments. For weekly assignments, a maximum of 48 hours will be allowed with penalty. There will be 25% penalty for each day, and no point is possible after 48 hours of the submission deadline.

Attendance / Participation Policy

The class attendance and discussion participations have up to 5% bonus in the final grade. The instructor is open to adjusting all things related to the class, if there is a good reason and justification.

Policy

Please email me if you face an unexpected situation that may impede your attendance, participation in required class and exam sessions, or timely completion of assignments.

Other Course Policies

Academic Integrity

As a student and member of the UIC community, you are expected to adhere to the Community Standards of academic integrity, accountability, and respect. Please review the UIC Student Disciplinary Policy for additional information.

You may not represent as your own work material that is transcribed or copied from another source, including persons, books, or Web pages. **Plagiarism is a serious violation of university policy and will not be tolerated.** You are welcome and encouraged to work together in learning the material. However, whatever you submit must be your own. In other words, cutting and pasting or copying verbatim from another

source be it a classmate, an online source or even something that the TA/instructor showed you is strictly forbidden.

The use of generative AI tools such as ChatGPT is permitted. *However, you should not use AI tools to ask for the exact questions.* In other words, students can be used to remind you of a concept or syntax of program statement, not to give you the exact solution**. Once an AI tool used and cited, the usage should be documented and disclosed with the submission before and after AI usage**. Students must cite any borrowed content sources to comply with all applicable citation guidelines, copyright law, and avoid plagiarism. All cases of suspected plagiarism will be reported to the Dean of Students for further review.

- **Cite Your Sources**: 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 clearly acknowledge 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, or your peers is certainly forbidden.
- **Inspiration is free**: you may discuss homework assignments with anyone. You are especially encouraged to discuss in online forum with your instructor and your classmates.
- **Plagiarism is forbidden**: the assignments and code that you turn in should be written entirely on your own. You should not need to consult sources beyond your textbook, class notes, posted lecture slides and notebooks, programming language documentation, and online sources for basic techniques. 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.
- **Do not search for a solution online**: You may not actively search for a solution to the problem from the internet or large language models. This includes posting to sources like StackExchange, Reddit, Chegg, etc. Searching/prompting for basic techniques in Python/Pandas/Numpy 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 or prompt "Here's the problem my professor 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.
- **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 the instructor. If it has not been described above, you should discuss it with us first

Please also pay attentions to the following netiquettes:

- Always consider audience. Remember that members of the class and the instructor will be reading any postings.
- Respect and courtesy must be provided to classmates and to instructor at all times. No harassment or inappropriate postings will be tolerated.
- Blackboard is not a public internet venue; all postings to it should be considered private and confidential. Whatever is posted on in these online spaces is intended for classmates and professor only. Please do not copy documents and paste them to a publicly accessible website, blog, or other space. If students wish to do so, they have the ethical obligation to first request the permission of the writer(s).

Email Expectations

Students are responsible for all information instructors send to your UIC email and Blackboard accounts. Faculty messages should be regularly monitored and read in a timely fashion.

IV. COURSE SCHEDULE

Week 1: Basic Introduction to the Course and Scala
Week 2: Recursion: Recursive Functions, Case classes, and Abstract Syntax Tree (Assignment#1, Quizz#1)
Week 3: Abstract Syntax Tree, inductive Definition and Attribute Grammars (Assignment#2, Quizz#2)
Week 4: Functional Operators, Big step semantics, Semantics for the Arithmetic Expressions (Quizz#3)
Week 5: Introduction to operational semantics, Intro to Lettuce, Let bindings (Spot Exam#1)
Week 6: Describe the syntax of Lettuce: a subset of ML language; Context-Free Grammar (Assignment#3, Quizz#4)
Week 7: Function Definition, Function Calls, and Recursion in Lettuce (Assignment#4, Quizz#5)
Week 8: Recursion in Lettuce; Syntax/Semantics for Explicit References (Spot Exam#2)
Week 9: Explicit References in Lettuce (cont.), Implicit References in Lettuce (Assignment#5, Quizz#6)
Week 10: Lettuce with Implicit References (Cont.), Calling Conventions (Quizz#7)
Spring Break
Week 11: Continuation Passing Style; Types in Lettuce, Type Annotated Program (Assignment#6, Quizz#8)
Week 12: Type Checking and Type inference in Lettuce (Spot Exam#3)
Week 13: Type Inference (cont.), Unification, and Polymorphism (Assignment#7, Quizz#9)
Week 14: Introduction to Logic Programming Languages (Assignment#8, Quizz#10)
Week 15: Introduction to Object Oriented Languages (Quizz#11)
Week 16: Final exam or final project week

Disclaimer

This syllabus is intended to give the student guidance on what may be covered during the semester and will be followed as closely as possible. However, as the instructor, I reserve the right to modify, supplement, and make changes as course needs arise. I will communicate such changes in advance through in-class announcements and in writing via Blackboard Announcements.

V. ACCOMMODATIONS

Disability Accommodation Procedures

UIC is committed to full inclusion and participation of people with disabilities in all aspects of university life. If you face or anticipate disability-related barriers while at UIC, please connect with the Disability Resource Center (DRC) at drc.uic.edu, via email at drc@uic.edu, or call (312) 413-2183 to create a plan for reasonable accommodations. To receive accommodations, you will need to disclose the disability to the DRC, complete an interactive registration process with the DRC, and provide me with a Letter of Accommodation (LOA). Upon receipt of an LOA, I will gladly work with you and the DRC to implement approved accommodations.

Religious Accommodations

Following campus policy, if you wish to observe religious holidays, you must notify me by the tenth day of the semester. If the religious holiday is observed on or before the tenth day of the semester, you must notify me at least five days before you will be absent. Please submit this form by email with the subject heading: **"YOUR NAME: Requesting Religious Accommodation."**

VI. CLASSROOM ENVIRONMENT

Inclusive Community

UIC values diversity and inclusion. Regardless of age, disability, ethnicity, race, gender, gender identity, sexual orientation, socioeconomic status, geographic background, religion, political ideology, language, or culture, we expect all members of this class to contribute to a respectful, welcoming, and inclusive environment for every other member of our class. If aspects of this course result in barriers to your inclusion, engagement, accurate assessment, or achievement, please notify me as soon as possible.

Name and Pronoun Use

If your name does not match the name on my class roster, please let me know as soon as possible. My pronouns are he/his/him. I welcome your pronouns if you would like to share them with me. For more information about pronouns, see this page: https://www.mypronouns.org/what-and-why.

Community Agreement/Classroom Conduct Policy

- Be present by turning off cell phones and removing yourself from other distractions.
- Be respectful of the learning space and community. For example, no side conversations or unnecessary disruptions.
- Use preferred names and gender pronouns.
- Assume goodwill in all interactions, even in disagreement.
- Facilitate dialogue and value the free and safe exchange of ideas.
- Try not to make assumptions, have an open mind, seek to understand, and not judge.
- Approach discussion, challenges, and different perspectives as an opportunity to "think out loud," learn something new, and understand the concepts or experiences that guide other people's thinking.
- Debate the concepts, not the person.
- Be gracious and open to change when your ideas, arguments, or positions do not work or are proven wrong.
- Be willing to work together and share helpful study strategies.
- Be mindful of one another's privacy, and do not invite outsiders into our classroom.

Content Notices and Trigger Warnings

Our classroom provides an open space for a critical and civil exchange of ideas, inclusive of a variety of perspectives and positions. Some readings and other content may expose you to ideas, subjects, or views that may challenge you, cause you discomfort, or recall past negative experiences or traumas. I intend to discuss all subjects with dignity and humanity, as well as with rigor and respect for scholarly inquiry. If you would like me to be aware of a specific topic of concern, please email or visit my Student Drop-In Hours.

VII. RESOURCES: Academic Success, Wellness, and Safety

We all need the help and the support of our UIC community. Please visit my **drop-in hours** for course consultation and other academic or research topics. For additional assistance, please contact your assigned college advisor and visit the support services available to all UIC students.

Academic Success

- UIC Tutoring Resources
- College of Engineering tutoring program
- Equity and Inclusion in Engineering Program
- UIC Library and UIC Library Research Guides.
- Offices supporting the UIC Undergraduate Experience and Academic Programs.
- Student Guide for Information Technology
- First-at-LAS Academic Success Program, focusing on LAS first-generation students.

Wellness

- **Counseling Services**: You may seek free and confidential services from the Counseling Center at https://counseling.uic.edu/.
- Access U&I Care Program for assistance with personal hardships.
- **Campus Advocacy Network**: Under Title IX, you have the right to an education that is free from any form of gender-based violence or discrimination. To make a report, email TitleIX@uic.edu. For more information or confidential victim services and advocacy, visit UIC's Campus Advocacy Network at http://can.uic.edu/.

Safety

- UIC Safe App—PLEASE DOWNLOAD FOR YOUR SAFETY!
- UIC Safety Tips and Resources
- Night Ride
- Emergency Communications: By dialing 5-5555 from a campus phone, you can summon the Police or Fire for any on-campus emergency. You may also set up the complete number, (312) 355-5555, on speed dial on your cell phone.