

CS4400/5400: Introduction to Programming Languages

Course Syllabus Spring 2024

Northeastern University

1 Logistics

Welcome to CS4400/5400! Please read this syllabus in its entirety.

- **Instructor:** Steven Holtzen `s.holtzen@northeastern.edu`
When emailing Steven or any of the TAs about course content, please include “CS4400” somewhere in the subject of your email.
- **Course time:** Monday/Wednesday 2:50PM – 4:30PM
- **Course location:** Hurtig Hall 224
- **Instructor Office Hours:** Thursday 11:00AM – 12:00PM in West Village H Room 310D
- **Teaching Assistants**
 - Minsung Cho `minsung@ccs.neu.edu`
 - Sam Stites `stites.s@northeastern.edu`
 - Jack Czenszak `czenszak.j@northeastern.edu`

See Canvas for teaching assistant office hours.

- **Input/Output:** Important messages will be sent as Canvas Announcements; you are responsible for ensuring that you receive these messages. We will have a course Piazza available here: <https://piazza.com/northeastern/spring2024/cs44005400>. Please ask questions on Piazza: you will likely receive faster answers there, and others can benefit from answers to your questions. Questions can be asked anonymously.

2 Course Overview

Our goal in this course is to study the essentials of programming languages: their purpose, their design, and their implementation. You have all programmed before, probably in multiple programming languages. You may have noticed differences and similarities between the programming languages and wondered: why are there so many languages? Why are they different and similar in these particular ways? Why is programming in one language sometimes easier than others? How (and why) are new programming languages made? We will study these questions bottom-up: we will *grow a tiny language* and add features to it iteratively, along the way encountering many of the core ingredients that go into today’s modern programming languages. We will also see some of the core *formal notions* that have been so effective in designing many of today’s languages: in particular, we will see formal language semantics and type systems. Finally,

we will go beyond our tiny language and explore some exotic languages that are radically different from ones you may have seen before. In the end, you should be equipped to learn new programming languages quickly, understand the key design decisions made by many of today's languages, and be prepared for a world where new programming languages are being made all the time.

2.1 Learning objectives

Module 0 – Functional programming in Plait Our main goal in this module is to become familiar with programming in Plait, the programming language that we will be using for our assignments and throughout the course. You probably haven't programmed in Plait before – this is by design! The language is intentionally small and designed specifically for this course, so you probably haven't heard of it before. It has important similarities with existing languages like OCaml and Racket, and is designed to be a minimal language for us to study some of the key concepts for the course, including functional programming, type systems, and macros.

Primary learning outcomes:

- Ability to translate an informal problem statement into a recursive solution in Plait.
- Ability to write Plait tests and specifications for your solutions.
- Knowledge of Plait's core features and capabilities.
- Ability to compare and contrast features of Plait with an existing language you are familiar with.

Module 1 – The Standard Model of Languages (SMoL) Our main goal in this module is to study the essence of a simple programming language with minimal features: this language is called the *Standard Model of Languages* (SMoL). To understand this language, we will write a program that runs SMoL programs (a.k.a., an *interpreter*): we will steadily add one feature to a time to the language and study its consequences. Along the way, we will learn how essential language features are implemented and how they interact.

Primary learning outcomes:

- Knowledge of the core language features of SMoL: syntax and semantics
- Know how to run an interpreter by hand
- Ability to implement and extend an interpreter for SMoL
- Know the difference between a core language feature and syntactic sugar
- Understanding of state and objects and how they relate to SMoL

Module 2 – Types Types constrain programs: they are a form of specification that tells us something about the kinds of values that programs accept as inputs and produce as outputs. Many of today's programming languages rely on types as a way to help programmers document the behavior of their code and *automatically* identify certain classes of errors. In this module, we will develop a type system for SMoL and use it as a vantage point for studying the modern landscape of today's type systems.

Primary learning outcomes:

- Be able to articulate what a type system is and why languages have them
- Familiarity with formal typing judgments and the ability run a typing derivation by hand
- Knowledge of type inference and ability to implement a type checker
- Understanding of subtyping and why it is useful
- Knowledge of important formal properties of type systems (safety and soundness)

Module 3 – Beyond the standard model Our exploration of SMoL can only take us so far: the world of programming languages is vast, and some language design decision are so radically different that they cannot be understood as "variations on a theme of SMoL". In this module we will explore languages with very different syntax and semantics from SMoL to see what programming looks like from these new perspectives.

Primary learning outcomes (time permitting):

- Understanding of logic-programming: syntax, semantics. Ability to implement a typechecker and evaluator for a small language in a logic programming setting.
- Understanding of lazy evaluation and how it differs from eager evaluation. Ability to implement an interpreter for a lazy language.
- Understanding of effects and effect handlers. Ability to add effect and handlers to SMoL. Knowledge of how effects are used in modern languages.

2.2 Course schedule

A detailed schedule of course topics and all assignment due dates is available at:

<https://docs.google.com/spreadsheets/d/e/2PACX-1vQNTDbNs-WnG7YU5iebhT9XuWfTNF2LBSPWzU1ctif8YrNuciQWZDtZu2hviaFt22asf1C2O27tdOoe/pubhtml?gid=0&single=true>

Be sure to check the schedule regularly for updates. Updates will be announced in class.

2.3 Textbook and reference materials

We will use the following open-access textbook: *Programming Languages: Application and Interpretation*, Third Edition, by Shriram Krishnamurthi. The textbook is available at:

<https://www.plai.org/>

You are encouraged to read the textbook ahead of class; the relevant sections to read are included in the course schedule. Some other good supplementary textbook resources for the course are:

- *Types and Programming Languages* by Benjamin Pierce.
- *Practical Foundations for Programming Languages (Second Edition)* by Robert Harper.
- *Essentials of Programming Languages (Third Edition)* by Daniel Friedman and Mitch Wand.

3 Prerequisites

This course will require that all students have the following background:

- Completed the course “CS3500: Object Oriented Design” or an equivalent course.
- Comfort programming in a major programming language.

If you have any questions about the prerequisites, please contact the instructor.

4 Academic Honesty

Cheating and other acts of academic dishonesty will be referred to Khoury College. There are very serious penalties here, so please do not take any chances by copying any material from the Internet or from other past or present students of this course or related courses. In particular, when completing the programming assignments, it is important that you do not refer to any completed solutions that you find on the Internet. When in doubt, ask the instructor or consult the Northeastern academic honesty page here: <http://www.northeastern.edu/osccr/academic-integrity-policy/>

5 Remote policy

The instructor will follow university policies on whether or not the course is to be taught in-person. The class is assumed by default to be in-person: the instructor will make an announcement if it will not be in-person. If the class is to be remote, it will be taught online using Zoom, and a link will be available in Canvas under the “Zoom” tab. Lectures may be recorded under certain circumstances. Please feel free to contact the instructor if you have any questions.

6 Coursework & Grading Policy

The course will consist of the following graded material, subject to change as the semester progresses (all changes to the grading scale will be announced as a Canvas announcement):

Type	Frequency	Percent of Final Grade
Homework	About once a week	60%
Quizzes	About 4	40%

- **Homeworks** are based on the previous week’s material. They may consist of programming assignments or written exercises. Programming assignments will primarily be done in Racket and Plait. Students may discuss the problems with other students, but must submit their own solutions. All problem sets and programming assignments will be turned in on Gradescope.
- **Quizzes** are non-cumulative and will cover all course content (including assignments). They will be taken online on Gradescope, and students will have 1 day to complete the quiz. *No collaboration is permitted on quizzes.*
- Some assignments, quizzes, and sub-problems will be marked as 5400 problems. These problems are designated for students enrolled in CS5400. Students enrolled in CS5400 are required to complete these exercises. They will not be graded for students enrolled in CS4400.
- The lowest homework grade will be dropped.

Late Work Policy All course assignments will be due at 11:59PM, and late work will be penalized according to the following scale:

- Less than 24 hours late: 5% penalty
- 24 – 48 hours late: 20% penalty
- More than 48 hours late: no longer accepted (at this point we will begin grading and require all work to be turned in).

This timeline is in place to ensure timely return of grades. If you require special accommodations or a grading extension, please email the instructor in advance of the deadline.

Grading thresholds Grades will be assigned according to the following standard letter-grade scale:

Range	> 93	[90,93)	[87, 90)	[83, 87)	[80, 83)	[77, 80)	[73, 77)	[70, 73)	[67, 70)	[60, 67)	<50
Grade	A	A-	B+	B	B-	C+	C	C-	D+	D	F

The instructor reserves the right to adjust final letter grades and the relative weights of assignments.

7 General resources

7.1 Disability Resource Center

Students who have disabilities who wish to receive academic services and/or accommodations should visit the Disability Resource Center <http://www.northeastern.edu/drc> at 20 Dodge Hall or call (617) 373-2675. Please be sure to provide your instructor with DRC's accommodations letter early in the semester in order to avoid logistical challenges. As-per the DRC's policy, it is the student's responsibility to coordinate with the instructor in order to provide suitable accommodations.

7.2 Title IX Notice

Title IX of the Education Amendments of 1972 protects individuals from sex or gender-based discrimination, including discrimination based on gender-identity, in educational programs and activities that receive federal financial assistance.

Northeastern's Title IX Policy prohibits Prohibited Offenses, which are defined as sexual harassment, sexual assault, relationship or domestic violence, and stalking. The Title IX Policy applies to the entire community, including students, faculty and staff of all gender identities.

Faculty members are considered "responsible employees" at Northeastern University, meaning they are required to report all allegations of sex or gender-based discrimination to the Title IX Coordinator.

If you or someone you know has been a survivor of a Prohibited Offense, confidential support and guidance can be found through University Health and Counseling Services staff <http://www.northeastern.edu/uahcs/> and the Center for Spiritual Dialogue and Service clergy members <http://www.northeastern.edu/spirituallife/>. By law, those employees are not required to report allegations of sex or gender-based discrimination to the University.

Alleged violations can be reported non-confidentially to the Title IX Coordinator within The Office for Gender Equity and Compliance at titleix@northeastern.edu and/or through NUPD (Emergency 617.373.3333; Non-Emergency 617.373.2121). Reporting Prohibited Offenses to NUPD does not commit the victim/affected party to future legal action.

In case of an emergency, please call 911.

Please visit <https://www.northeastern.edu/ouec> for a complete list of reporting options and resources both on- and off-campus.