

# SI 413: Programming Languages and Implementation

## Course Policy, Fall AY2024

### Instructors:

- Prof. Daniel S. Roche, 438 Hopper Hall, x36814, [roche@usna.edu](mailto:roche@usna.edu) (Coordinator).
- Dr. Keith Sullivan, 453 Hopper Hall, x36810, [ksulliv@usna.edu](mailto:ksulliv@usna.edu).

### MGSP Leader:

- 1/C Michael Huizenga, 19<sup>th</sup> company, [m242892@usna.edu](mailto:m242892@usna.edu)

Course Description: This course examines basic concepts underlying the design of modern programming languages: types, control structures, abstraction mechanisms, inheritance, concurrency and constructs for programming. The course includes programming assignments in several languages.

Credits: 2-2-3

### Learning Objectives:

1. Understand the functional programming paradigm and be able to solve problems in a functional language (supports outcome CS-6).
2. Develop a vocabulary for describing programming languages (supports outcome CS-6).
3. Understand lexical analysis, parsing, and basic interpretation (supports outcome CS-6).
4. Implement a simple interpreter (supports outcome CS-6).
5. Learn a new programming language independently and use it to solve basic tasks (supports outcome CS-6).

Student Outcomes: Graduates of the program will have an ability to:

- 1. Analysis.** Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
- 2. Implementation.** Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
- 3. Communication.** Communicate effectively in a variety of professional contexts.
- 4. Ethics.** Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
- 5. Teamwork.** Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
- CS-6. Theory.** Apply computer science theory and software development fundamentals to produce computing-based solutions.
- IT-6. Requirements.** Identify and analyze user needs and to take them into account in the selection, creation, integration, evaluation, and administration of computing based systems.

## Syllabus:

1. **Unit 1: Programming Language Basics** (Classes 1–2)  
Course overview, Classifying programming languages, Steps of compilation
2. **Unit 2: Scheme Basics** (Classes 3–5)  
Lists, Evaluation, Functions, Symbols, Let
3. **Unit 3: Advanced Scheme** (Classes 6–8)  
Lambda, Side effects, Mutation, Tail recursion
4. **Unit 4: Scanning and Parsing** (Classes 9–13)  
Scanning, Top-down parsing, Recursive descent, Shift-reduce, CFSMs, SLR
5. **Unit 5: Semantic Analysis** (Classes 14–15)  
Abstract Syntax Trees, Static Type Checking
6. **Unit 6: Names and scopes** (Classes 16–18)  
Lifetimes, Nested scopes, Declaration order, Function calls, Lexical scope, Closures
7. **Unit 7: Functions** (Classes 19–20)  
Parameters, Polymorphism, Macros, Lazy evaluation, Built-ins
8. **Unit 8: Implementation issues** (Classes 21–24)  
Garbage collection, Intermediate Representations, Optimization, Virtual Machines
9. **Unit 9: Control** (Classes 25–27)  
For loops, iterators, generators, go-to, generics

Updates to the course policy: In case this course policy needs to be changed during the semester, students will be notified by email and verbally during class. The current version will always be posted on the course website.

## Textbooks:

- Michael L. Scott. *Programming Language Pragmatics*, Morgan Kaufmann Publishers. Optional but recommended. The current edition is the 4th (2015), but any edition is OK. Page numbers in the notes are from the 3rd edition.
- Abelson, Sussman, and Sussman. *Structure and Interpretation of Computer Programs*, MIT Press. Out of print but available used or [for free online](#).
- Aho, Lam, Sethi, and Ullman. *Compilers: Principles, Techniques, and Tools*, Pearson, 2006. Completely optional, recommended for those wishing to gain deeper knowledge of compilers.

Course Website: The primary website for course content is

<https://faculty.cs.usna.edu/~roche/413/>

This site is only available on the Yard, on the USNA intranet. To access exactly the same content from anywhere else, you can also use the following unofficial mirror:

<https://www.roche.work/413/>

Extra Instruction: Extra instruction (EI) is strongly encouraged and should be scheduled by email. (For Dr. Roche, [first go here to check available times.](#)) EI is not a substitute lecture; students should come prepared with specific questions or problems.

Collaboration: The guidance in the Honor Concept of the Brigade of Midshipmen and the Computer Science Department Honor Policy must be followed at all times. See <https://www.usna.edu/CS/resources/honor.php>. Specific instructions for this course:

- Collaboration or assistance from any human other than the instructors and those enrolled in SI413 this semester is not permitted. This includes any written or electronic materials from previous semesters.
- Homework: Students may collaborate on homework with others in the same class, but must cite this collaboration clearly. Every student must actually complete their own assignment and understand anything they turn in.
- Labs: Labs represent the main work of this class. Most labs will be completed in groups (pairs). Students in the same group for that lab may discuss and collaborate freely with each other, and only need to turn in one combined submission. Informal discussion between students in other groups is allowed, but *each group must do their own work, and may not share any specific (even partial) solutions*. Generally, for labs, any discussion between groups should be done away from a computer, and no student should turn in any code that another student has seen.
- Projects: Projects will be completed in groups (pairs). Students may discuss any aspect with classmates who are working on different languages (topics), but may never discuss or collaborate with other groups working on the same language (topic).
- Exams: No collaboration is allowed. Each student may prepare and bring a single study sheet to the midterm and final exams, but these must be prepared individually (no photocopies). Any group study guides should be shared with the instructor.

All collaboration and outside sources should always be cited. The same rules apply for giving and receiving assistance. If you are unsure whether a certain kind of assistance or collaboration is permitted, you should assume it is not, work individually, and seek clarification from your instructor.

Use of Generative AI: The use of generative AI tools to help complete assignments is treated the same as collaboration or assistance with a human (see above) and is therefore prohibited under most circumstances. Please talk with your instructor if you believe there are ways to use generative AI tools without hindering the course learning objectives.

#### Classroom Conduct:

Everyone in the classroom will show appropriate respect to each other at all times.

The section leader is responsible for recording attendance, bringing the class to attention, notifying the CS department office if the instructor is more than 5 minutes late, and directing the class in useful work in the instructor's absence.

Drinks are permitted, but they must be in closable containers. Food, alcohol, and tobacco (of all kinds) are prohibited. The use of laptops, tablets, or other devices during class is at the discretion of the instructor; such use must be related to the class and should never serve as a distraction to other students.

Absences:

Students are responsible for all class material. Notes will be posted for each lecture, along with recommended readings. However, this material is not exhaustive and students missing class should arrange to copy notes from a classmate.

Late Policy: Because solutions will generally be posted or discussed immediately, **no late work will be accepted for a grade**. Even in cases of excused absences, all work must be submitted by the posted deadline. Exceptions to this rule are possible under exceptional circumstances, at the discretion of the instructor.

Grading:

Your final grade will be computed as follows:

- 10%: Weekly homeworks
- 30%: Labs
- 17%: Course project
- 8%: Scheme Practicum
- 10%: Midterm exam
- 25%: Final exam

Plus/minus grades will be assigned based on the following numerical cutoffs:

	-		+	
<b>A</b>	90–92	93–100		
<b>B</b>	80–82	83–86	87–89	
<b>C</b>	70–72	73–76	77–79	
<b>D</b>		60–66	67–69	
<b>F</b>		0–59		

Here is a breakdown of percentages by grading period, according to the anticipated schedule of due dates:

	6 weeks	12 weeks	16 weeks	Final
Homework	20%	15%	13%	10%
Labs	44%	41%	40%	30%
Projects	0%	7%	23%	17%
Exams	36%	37%	24%	43%
Total	100%	100%	100%	100%

Submitted:

---

Assoc. Prof. Daniel S. Roche  
Course Coordinator