

Syllabus

Objectives

1. Learn new ways of thinking about problems and programs.
2. Make it easier to learn languages, by learning principles and concepts that apply to many languages.
3. Learn some interesting languages.

Prerequisite

- CS 205, CS 211
- Java; the memory model of C, including pointers; predicate calculus

Books

Recommended: Michael L. Scott, *Programming Language Pragmatics*, 4th Edition

The textbook does not include all material that will be covered in class.

Grading

There will be two midterm exams and a final exam, as well as programming projects and graded homework. The bulk of your grade will be determined by the exams, but doing the assignments will be key to learning the material and doing well on the exams. You can expect the exams to base questions on the assignments.

The breakdown is:

- Assignments: 25%
- Midterm exam 1: 20%
- Midterm exam 2: 20%
- Final Exam: 35%

Collaboration

All students are expected to be familiar with and abide by the DCS academic integrity policy, found at <http://www.cs.rutgers.edu/policies/academicintegrity/>. Any work you submit must be your own work, and must not include material obtained from other students, books, or any sources on-line.

Students are encouraged to discuss projects with each other, but must not share code. After discussion, we recommend that students discard any written material and wait an hour or so before resuming work on the project. The goal is for you to submit work that you understand, and not to submit rote copies of other students' material.

Assignments

Assignments will be submitted through Sakai. Deadlines have deliberately been placed very early in the morning, to discourage working to the last possible minute. The worst possible time for a computer problem or other emergency to happen is right before a submission deadline, so make an effort not to find yourself in that situation.

Assignment deadlines are strictly enforced. Once the deadline passes, no submissions will be accepted for any

reason, unless explicit arrangements have been made in advance. Students are advised to plan ahead and complete work ahead of schedule.

Topics

We expect to cover this material during the semester, time permitting. Note that it is arranged by topic, and not chronologically.

1. Formal languages
 - Context-free grammars
 - Regular grammars
 - Finite-state automata
 - Regular expressions
 - Using all of the above to define a language
2. Functional programming
 - Haskell
 - Repetition through recursion, not iteration
 - Stateless programming
 - Closures
3. Logical programming
 - Prolog
4. Scripting / "Dynamic" programming
 - Python
 - Schell scripting
5. Principles
 - Types and type inference
 - Scope and binding, lexical and dynamic
 - Parameter passing modes