

Spring 2020

- Home
- Announcements
- Syllabus
- Modules
- Assignments
- Quizzes
- Discussions
- KSL Research Guides
- Purchase from the Bookstore

## Course Syllabus

[Jump to Today](#)

## Description and Objectives

### Course Description

This course examines the four main programming paradigms: imperative, object-oriented, functional, and logical. It is assumed that students will come to the course with significant exposure to object-oriented programming and some exposure to imperative programming. The course will teach the functional paradigm in depth, enhance the students' knowledge of the object-oriented and imperative paradigms, and introduce the logical paradigm. The course will explore language syntax, semantics, names/scopes, types, expressions, assignment, subprograms, abstraction and inheritance. This exploration will have several forms. Students will study the programming language concepts at a theoretical level, use the concepts in functional language programming, and implement the concepts by designing language interpreters.

### Learning Objectives

1. Become fluent in multiple programming paradigms, in particular the functional paradigm.
2. Understand, at a deep level, the different components that make up a programming language.
3. Be able to reason mathematically about the semantics of a program.

#### Pre-requisites

- EECS 233: Data Structures
- EECS 302: Discrete Mathematics

## Course Materials

#### Required Textbooks:

- Scott, *Programming Language Pragmatics*, 4th edition, 2016. ISBN 9780124104099
- Friedman and Felleisen, *The Little Schemer*, 4th edition, 1996. ISBN 9780262560993

**Language Tools and Environments:** You will need to load the different language environments we will be using onto your personal computer. A full description is posted in the class Modules.

## Resources

**Lectures:** Monday, Wednesday, Friday at 3:20pm in Nord 410

**Instructor Office Hours:** Monday 4:30-5:30pm, Tuesday 3:00-4:00pm, Wednesday 10:30-11:30am, Thursday 9:00-10:00am at <https://cwru.zoom.us/j/986854791>.

**Teaching Assistant Office Hours:** Monday 10:30-11:30 in the Glennan 3rd floor student lounge, Tuesday 1-2 in Glennan 313, and Thursday 5:30-6:30 in the Glennan student lounge.

## Assessments

#### Late Work Policy:

All work is expected to be submitted on time and a significant number of points will be deducted on late work. The only exceptions are for medical or similar emergencies, and in such situations work will only be accepted with notice from an appropriate university official. **If you are unable to complete assigned work on time for any (valid) reason, contact your navigator, and your navigator will collect the needed documentation and coordinate with me on either an extension, alternate work, or excusing the work.** *If you have to miss class due to a scheduled event such as a sporting event or club performance, you must make arrangements to submit your required work early.*

The following late penalties will be used.

For programming projects and exercises: -5% for 1 hour late, -10% for up to 12 hours late, -25% for up to 48 hours late, -50% more than 48 hours late

For written homework: -10% if up to 48 hours late, -25% if up to 72 hours late, -50% if more than 72 hours late

No work will be accepted after solutions are given, either in lecture or posted on Canvas, and no work will be accepted after April 27.

#### Written Exercises:

There will be occasional written exercises throughout the term. The purpose of these exercises is to strengthen your understanding of the techniques discussed in class and to help you implement these techniques in your projects. Each student is expected to work on the written exercises on their own. You may work with another student on the exercises, *but you must write up your solutions without consulting any work you did together.*

#### Programming Exercises:

There will be a few programming exercises in the course. The purpose of the exercise is to gain experience with a specific programming language or paradigm. Each student is expected to work on the exercises on their own, but students may discuss general techniques or help debug errors. There should be no electronic copying of code.

#### Programming Projects:

The goal of the programming projects is to gain a deep understanding of the theory of programming languages by building an interpreter for different language features. Because the of the challenging nature of the programming projects, you may work in a team of up to three students in the course to solve the problems. This collaboration may include sharing a code repository, splitting up tasks of the assignment, etc. However, each student is still expected to submit their own solutions and **each student is responsible for understanding the code submitted in their name.** *You may not receive direct coding assistance from any student who has previously taken the course.*

#### Midterm:

The midterm will be in class on Monday, February 24.

#### Final:

The final exam is at the time specified by the registrar.

#### Calculators:

*No calculators will be permitted on midterms and finals.*

#### Grading Scheme:

|                        |     |
|------------------------|-----|
| Written Exercises:     | 10% |
| Programming Exercises: | 10% |
| Programming Projects:  | 30% |
| Midterm:               | 15% |
| Final Exam:            | 35% |

#### Grade scale:

The class will use the following scale to translate your percentage grade to a final grade.

| If your percentage grade is at least | Your final grade will be at least |
|--------------------------------------|-----------------------------------|
| 90                                   | A                                 |
| 80                                   | B                                 |
| 70                                   | C                                 |
| 60                                   | D                                 |

*You must score at least 50% on the final exam to pass the course.*

## Academic Honesty















Please see the general [University Policy on Academic Integrity](#) <sup>↗</sup>. The specifics for the course are listed above in the Assessments section.

## Special Considerations

**Physical Disabilities or Other Hardships:** If you have a physical disability or other hardship that can potentially put you at a disadvantage in this course, please see [Disability Resources](#) <sup>↗</sup>. They will make certain you receive the necessary accommodations so that you may perform your best.

**Religious Holidays:** I strive to schedule all major projects and tests so that they do not conflict with important religious holidays. However, I am not always successful in doing that. If an important religious holiday conflicts with a class test or assignment in a way that makes it so that you can not take the test or complete the assignment as originally assigned, please see me as soon as possible to make necessary arrangements.

## Course Summary:

| Date             | Details  |
|------------------|--|
| Wed Jan 29, 2020 |  <a href="#">Programming Exercise 1</a> due by 11:59pm  |
| Fri Feb 7, 2020  |  <a href="#">Written Exercise 1</a> due by 3:20pm   |
| Mon Feb 17, 2020 |  <a href="#">Interpreter.Part 1</a> due by 11:59pm  |
| Mon Feb 24, 2020 |  <a href="#">Midterm</a> due by 3:20pm  |
| Wed Feb 26, 2020 |  <a href="#">Programming Exercise 2</a> due by 11:59pm  |
| Mon Mar 23, 2020 |  <a href="#">Interpreter.Part 2</a> due by 2pm<br> <a href="#">Programming Project.Part 2.5</a> due by 11:59pm |
| Wed Apr 1, 2020  |  <a href="#">Interpreter.Part 3</a> due by 11:59pm  |
| Fri Apr 10, 2020 |  <a href="#">Written Exercise 2</a> due by 3:20pm   |
| Fri Apr 17, 2020 |  <a href="#">Programming Exercise 3</a> due by 11:59pm  |
| Mon Apr 27, 2020 |  <a href="#">Interpreter.Part 4</a> due by 11:59pm  |
| Wed Apr 29, 2020 |  <a href="#">Final Class Exercise</a> due by 11:59pm  |
| Wed May 6, 2020  |  <a href="#">Final Exam</a> due by 3pm  |
| Thu May 7, 2020  |  <a href="#">Course Evaluation</a> due by 11:59pm   |

| May 2020 |    |    |    |    |    |    |
|----------|----|----|----|----|----|----|
| <        |    |    |    |    |    | >  |
| 26       | 27 | 28 | 29 | 30 | 1  | 2  |
| 3        | 4  | 5  | 6  | 7  | 8  | 9  |
| 10       | 11 | 12 | 13 | 14 | 15 | 16 |
| 17       | 18 | 19 | 20 | 21 | 22 | 23 |
| 24       | 25 | 26 | 27 | 28 | 29 | 30 |
| 31       | 1  | 2  | 3  | 4  | 5  | 6  |

| Assignments are weighted by group: |        |
|------------------------------------|--------|
| Group                              | Weight |
| Programming Exercises              | 10%    |
| Written Exercises                  | 10%    |
| Programming Projects               | 30%    |
| Tests                              | 15%    |
| Extra Credit                       | 0%     |
| Final Exam                         | 35%    |
| Total                              | 100%   |