Course Goals
The course aims to teach students about the fundamentals of programming languages as well as provide experience in functional and object-oriented programming. This will include lectures on various programming language topics (e.g., type systems) and analyzing numerous existing programming languages to understand their design goals. Students will also learn a subset of the Racket, Haskell, and Java languages to apply the topics from the lectures.

Course Structure
The course content includes lectures, in-class programming sessions, programming assignments as homework, quizzes, and exams. An example week of class will include: lecture on Tuesday, an in-class programming session on Thursday that covers specific features of a particular programming language, and a quiz on the weeks lecture. Homeworks will be programming assignments using a particular programming language every 2-3 weeks. Exams will include traditional exam questions as well as programming problems.

Topics
- Paradigms of programming languages
- Functional programming
- Object-oriented programming
- Type systems
- Algebraic data types
- User-defined types
- Polymorphism
- Garbage collection algorithms
- Racket programming language
- Haskell programming language
• Java programming language

**Lectures & in-class programming sessions**
The majority of class time will be used for lectures or in-class programming sessions. The lecture slides will be made available on the course website. Students are expected to bring laptops to all classes and be able to install the necessary software. Attendance will not be explicitly recorded for a grade, but students are expected to follow along to the lectures and participate in the programming sessions.

**Quizzes**
Quizzes will be frequently given in-class that cover the lecture and homework material. These will be taken in-class through Canvas on students’ own laptops. Quizzes cannot be made up if the student is absent (and may not be taken outside of class). A quiz grade may be excused if a documented excuse is provided prior to the quiz. No materials (e.g., slides, website, a neighbor) may be used while taking quizzes.

**Exams**
A midterm and final exam will be taken through Canvas on students’ own laptops. Exams will cover material from lectures and homework material, including traditional exam-style questions (e.g., fill in the blank or multiple choice) as well as programming problems. Students will be provided further instructions on what may be used during exams (e.g., slides or Google).

**Homeworks**
Homeworks will be assigned throughout the semester that require students to turn in code to Canvas using a specific programming language and set of libraries. Instructions must be followed (e.g., the name of the submitted file) and programs must execute without modification. Expected outputs will be provided. Sharing code will be considered as plagiarism. Turning in homework late will result in a 10% point deduction per day late. One day late begins 1 minute after the deadline and extends to 23 hour 59 minutes after the deadline, and this patterns repeats for N days late.

**Grading**
- 40% programming homework
  - 6 assignments are anticipated
- 40% quizzes
  - At least 10 quizzes are anticipated
- 20% exams
  - Midterm and final
I will use the standard grading scale to convert to letter grades with plus and minus.

The instructor reserves the right to revise, alter or amend this syllabus as necessary. Students will be notified regarding any changes.