

COSE212: Programming Languages

Instructor: Hakjoo Oh

Fall, 2019

Basic Information

- Instructor: Hakjoo Oh
 - Position: Associate professor in Computer Science and Engineering, Korea University
 - Expertise: Programming Languages and Software Analysis
 - Office: 616c, Science Library
 - Email: hakjoo_oh@korea.ac.kr
 - Office Hours: 1:00pm–3:00pm Mondays and Wednesdays (by appointment)
- Course Website:
 - <http://prl.korea.ac.kr/~pronto/home/courses/cose212/2019>
 - Course materials will be available here (not on Blackboard).

Objectives The primary goal of this course is to teach essential concepts of programming languages. In this course, you will learn the concepts by designing and implementing interpreters of programming languages. Topics include:

- **Part 1 (Preliminaries):** inductive definition, functional programming
- **Part 2 (Basic concepts):** syntax, semantics, naming, binding, scoping, environment, interpreters, states, side-effects, store, reference, mutable variables, parameter passing
- **Part 3 (Advanced concepts):** type system, typing rules, type checking, soundness/completeness, type inference, polymorphism, modules, module procedures, typed modules, objects, classes, methods, inheritance, typed object-oriented languages

The secondary goal of this course is to familiarize students with functional programming. Functional programming, which encourages using pure functions rather than making side effects, is one of the major programming paradigms and often found in modern programming languages such as Python, JavaScript, C++, Java8, Scala, Go, etc. In this course, you will learn functional programming with OCaml¹ and use it to implement the interpreters.

Prerequisites This is not an introductory course on computer programming. You should have a strong understanding of the basic computer science courses such as C/Java programming, theory of computation, discrete mathematics, and data structures.

Textbook:

- Self-contained slides will be provided.
- Reference:
 - Essential concepts of programming languages (third edition). Daniel P. Friedman and Mitchell Wand.

¹<https://ocaml.org>

Grading (tentative):

- Homework – (50–70%)
 - 5–7 programming assignments in OCaml
- Final exam – (25–45)%
- Attendance – 5%

Assignment policy:

- No late submissions will be accepted.
- All homework assignments must be your own work. **Copying assignments will get you 0 point for the entire HW score.**

Schedule (tentative):

Weeks	Topics
Week 1	Introduction
Week 2	Inductive Definition
Week 3	Functional Programming
Week 4	Functional programming
Week 5	Expressions
Week 6	Procedures
Week 7	Procedures
Week 8	States
Week 9	Mid-term exam (no class)
Week 10	Simple Type System
Week 11	Simple Type System
Week 12	Type Inference
Week 13	Type Inference
Week 14	Polymorphic Type System
Week 15	Lambda Calculus
Week 16	Final exam