COSE212: Programming Languages

Lecture 0 — Course Overview

Hakjoo Oh
2026 Fall

Basic Information

Instructor: Hakjoo Oh

- **Position:** Professor in CS, Korea University
- **Expertise:** Programming Languages, Software Engineering
- **Office:** 505, Jung Woonoh IT & General Education Center
- **Email:** `hakjoo_oh@korea.ac.kr`
- **Office Hours:** 1:00pm–2:00pm Mondays (by appointment)

TAs:

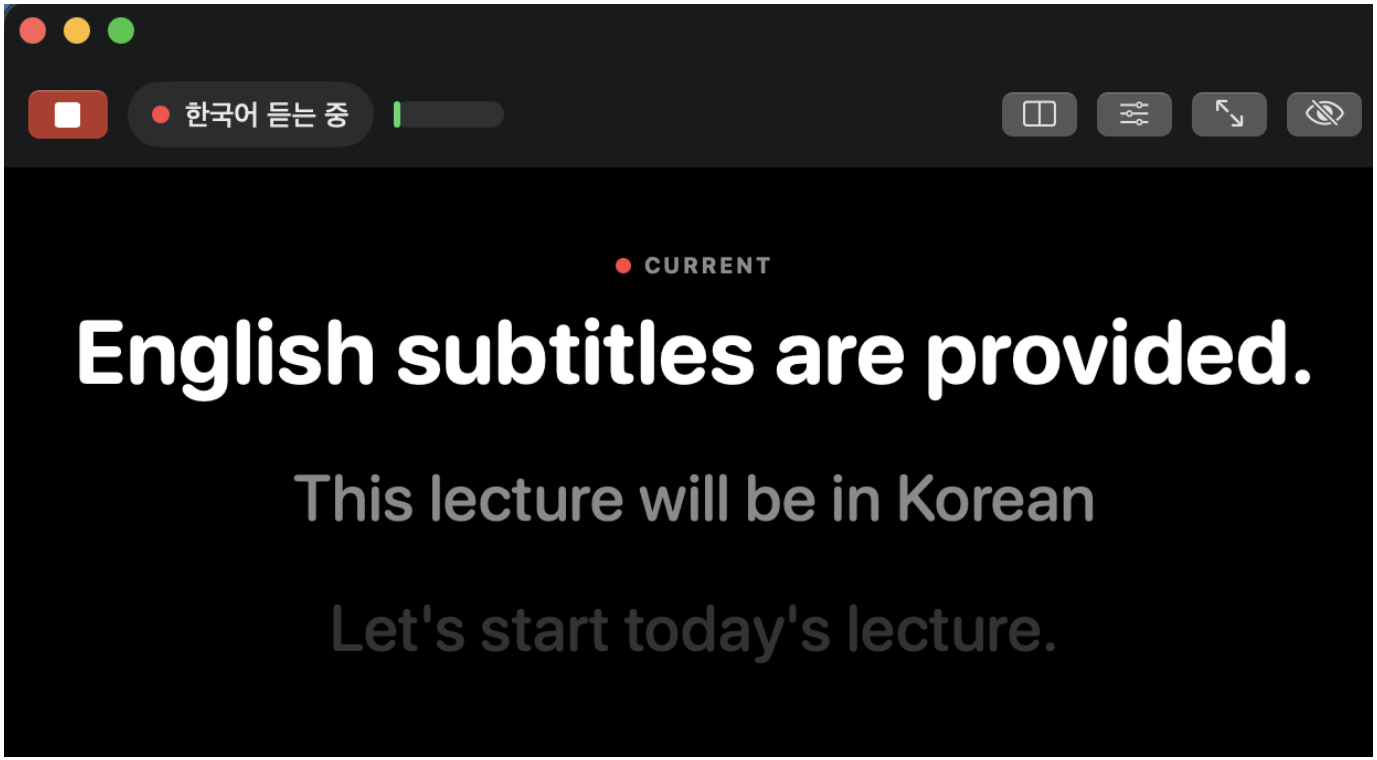
- Mingyu Jo (mingyu_jo@korea.ac.kr)

Course Website:

- `https://prl.korea.ac.kr/courses/cose212/2026/`
- Course materials will be available here.

Course Language

Korean with English subtitles



Vibe Coding

한국어 강의에서 실시간으로 영어 자막을 생성하고 화면에 보여주는 프로그램 만들어줘. 인터넷 연결없이 로컬에서 돌아갈 수 있어야 하고, 번역 품질이 최대한 좋아야 해.

3분 30초 동안 작업 중입니다

먼저 현재 프로젝트 구조와 실행 환경을 확인한 뒤, 완전 오프라인 음성 인식·번역·자막 표시 파이프라인으로 구현하겠습니다. 번역 품질을 우선해 로컬 모델을 기본으로 하되, 하드웨어에 따라 경량 설정으로도 전환할 수 있게 만들고 실제 실행까지 검증하겠습니다.

명령을 실행했습니다

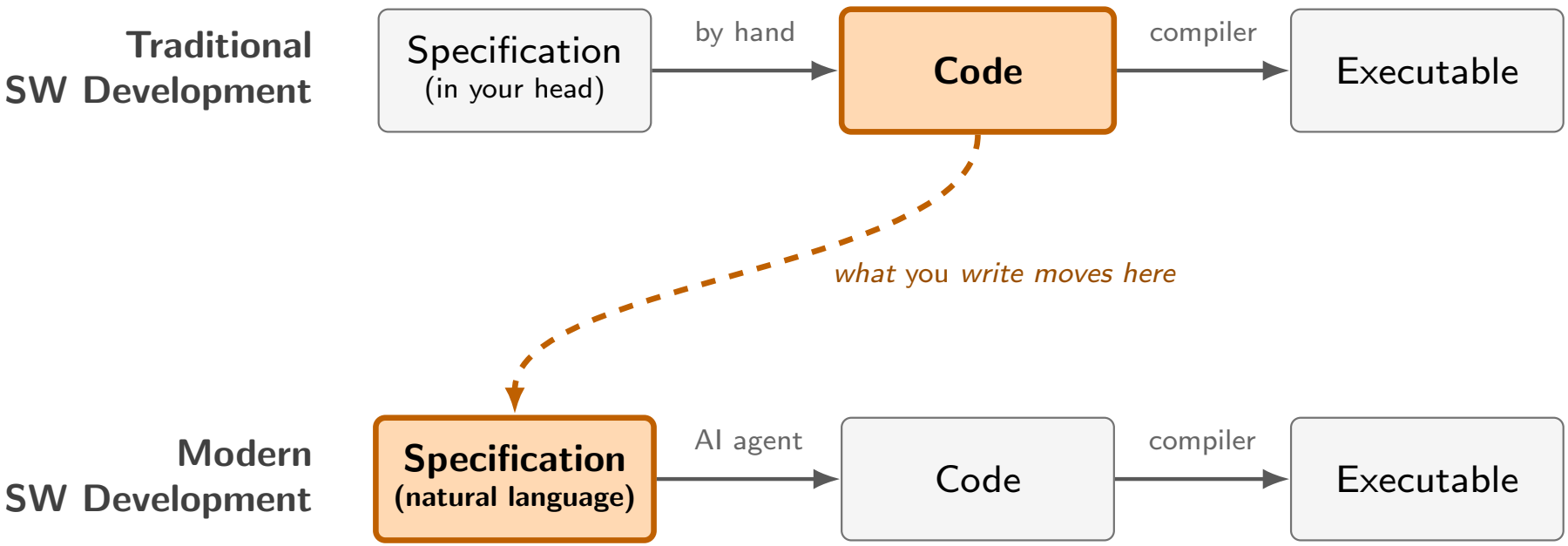
프로젝트가 비어 있어 macOS Apple Silicon용 데스크톱 앱으로 새로 구성하겠습니다. 마이크 입력은 짧게 겹쳐 처리해 문장 누락을 줄이고, 1단계에서 한국어 음성을 정확히 받아쓴 뒤 2단계 로컬 번역 모델로 영어를 만드는 구조를 택하겠습니다. 최초 모델 설치 이후에는 네트워크를 완전히 끊어도 실행되도록 캐시 검사와 오프라인 모드를 강제하겠습니다.

명령을 실행했습니다 웹 검색함

기본 품질 모드는 이 Mac의 넉넉한 메모리를 활용해 Whisper large-v3와 TranslateGemma 12B를 함께 사용하도록 잡았습니다. Whisper의 음성→영어 직역에만 의존하지 않고, 한국어 원문을 먼저 확정한 뒤 번역 전용 모델을 거치므로 전문 강의 문맥과 문장 자연스러움이 더 좋습니다. 실시간성을 더 중시할 때는 더 작은 모델 조합도 설치할 수 있게 구성하고 있습니다.

Designing model profiles and download management

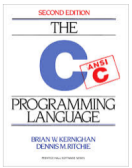
SW Development in the Age of AI Agents

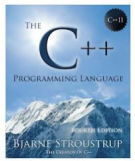


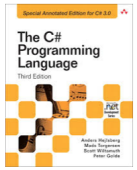
About This Course

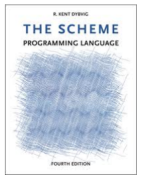
This is not an introductory course on programming.

- You will not learn particular programming languages










- You will not learn how to write programs in those languages

Instead, in this course you will learn

- how programming languages are designed and implemented
- fundamental principles of modern programming languages
- thinking formally and rigorously

To succeed in this course, you must

- have basic programming skills
- be familiar with at least two PLs (e.g., C, Java)
- have taken Theory of Computation, Discrete Math, etc
- be prepared to learn new things

Design and Implementation of Programming Languages

You will learn programming language concepts by designing and implementing our own programming language system.

- We will define a programming language. For example, "factorial" is written in our language as follows:

```
let x = read in
letrec fact(n) =
  if iszero n then 1
  else ((fact (n-1)) * n)
in (fact x)
```

- We will design and implement an interpreter for the language:

$$\text{Program} \longrightarrow \boxed{\text{Interpreter}} \longrightarrow \text{Result}$$

- We will design and implement a type checker for the language:

$$\text{Program} \longrightarrow \boxed{\text{Type Checker}} \longrightarrow \text{Safe/Unsafe}$$

Functional Programming

The secondary goal of this course is to be familiarized with functional programming, which encourages using pure functions rather than making side effects.

- Functional programming is one of the major programming paradigms adopted in modern programming languages such as Python, JavaScript, C++, Java8, Scala, Go, etc.
- In this course, you will learn functional programming with OCaml[1] and use it to implement programming languages.

[1] `https://ocaml.org`

Topics

- **Part 1 (Preliminaries):** inductive definition, basics of functional programming, recursive and higher-order programming
- **Part 2 (Basic concepts):** syntax, semantics, naming, binding, scoping, environment, interpreters, states, side-effects, store, reference, mutable variables, parameter passing, exceptions, classes
- **Part 3 (Advanced concepts):** type system, typing rules, type checking, soundness/completeness, automatic type inference, polymorphic type system, lambda calculus

Course Materials

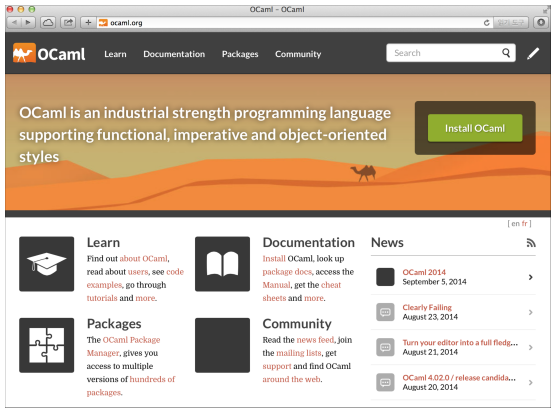
- Self-contained slides will be provided.
  - You are required to attend every class (otherwise, it'd be difficult to catch up)
- Hakjoo Oh. Introduction to Principles of Programming Languages.
  - Korean version
  - English version (AI-translated)

Grading (changed)

- Programming assignments – 0%
  - 4 programming assignments
- Final exam – 80%
- Attendance & Participation – 20%

Programming in ML

- ML is a general-purpose programming language, reflecting the core research achievements in the field of programming languages.
  - higher-order functions
  - static typing and automatic type inference
  - parametric polymorphism
  - algebraic data types and pattern matching
  - automatic garbage collection
- ML inspired the design of modern programming languages.
  - C#, F#, Scala, Java, JavaScript, Haskell, Rust, etc
- We use OCaml, a French dialect of ML:



`http://ocaml.org`

Questions?