COSE212: Programming Languages

Lecture 18 — Course Review

Hakjoo Oh
2018 Fall

About This Course (from Lecture 0)

This course is *not* about

- to learn particular programming languages

THE C PROGRAMMING LANGUAGE
THE C++ PROGRAMMING LANGUAGE
Java
The C# Programming Language
python
JavaScript
THE SCHEME PROGRAMMING LANGUAGE
php

- to improve your "programming skills" (e.g., tools, libraries, etc)

Instead, in this course you will learn

- fundamental principles of modern programming languages
- how programming systems are designed and implemented
- 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 (from Lecture 0)

We 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:

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

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

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

Checklist

Have you pick up the following ideas from this course?

- Designing programming languages (i.e., syntax and semantics)
- Implementing programming languages (i.e., interpreters)
- Detecting runtime errors at compile-time (i.e., type system)

Applications of Programming Language Foundations

A good understanding of programming language foundations is essential for advanced study of software:

- Software engineering
- Software security
- Software analysis
- ...

한 학기 수고 많았습니다!