



L1: Programming Languages

▼ 유형	강의
☑ 복습 여부	☑
⚙ Status	Done

What is PLT?

Learning the fundamental **principles underlying programming language design** by **implementing language features** using a combination of interpreters and little compilers.

- how are the programming languages we use everyday designed ?

Course Objective of PLT

- having the ability to choose the most appropriate programming model and language for a given problem
- learning any programming languages quickly
- learning fundamentals to design a light programming language
 - design your own programming languages for your specific projects and be ready to learn compiler theory

▼ learn open-source software development skills

[Use if IDE and OSS platform] Students can use the Racket IDE tool to implement simple functional programs.

[Use of APIs] Students can read API documents and use them in implementing programming tasks.

[OSS developer] Students can create a pull request to any programming language interpreter.

All the programming is done in Racket, a multi-paradigm programming language.

Why do you want to take a PL class?

- I heard that this course helps students learn a new language quickly. I find it challenging to grasp new programming languages quickly and I hope this course can help me improve my weakness in this regard.

What are languages?

Language is a complex system that incorporates words, structure, and grammar to facilitate communication. It is also a tool for expressing and organizing one's thoughts to ourselves and other people.

What are Programming languages?

Programming languages serve as a means to communicate with computers. With the help of programming languages, we can tell computers what tasks to perform, how to execute them, and in what order. Learning programming languages teach us how to analyze problems, design algorithms, and implement solutions.

What do languages consist of? (Say anything in various aspects.)

Language consists of various components.

For instance, language includes elements such as alphabet, words, syntax, idioms, sentences, phrases, semantics, and so on.

From a linguistic perspective, the major components of language are phonology(sound), morphology(words), syntax(sentence structure), semantics(meaning), and pragmatics(language usage in a context).

What do Programming languages consist of?

Programming language is composed of various components. For instance, programming languages consist of keywords, syntax, semantics, variables, functions, operators, expressions, control flow, and so on. Both programming languages and natural languages share a common interest in syntax and semantics. Linguists explore these aspects in natural languages, while Programming Language Theory analyzes the syntax and semantics of programming languages.

Are they important to choose a proper PL for your project? Why?

When considering the supported devices for one's project, choosing the appropriate programming language becomes crucial. This is because programming languages vary in their compatibility with different supported devices. This compatibility is influenced by factors such as the underlying architecture of the devices, the operating systems they run, and the available libraries. For instance, Swift is primarily used for developing applications on Apple devices. Java is frequently used in developing web applications and Android apps, as long as the Java Virtual Machine(JVM) is accessible for the given platform. Moreover, individual systems are rarely built using just one programming language. Instead, different programming languages are used for various components within a system. For example, a web application may include database queries written in SQL, server-side code written in Java, and client-side code written in React.

Therefore to choose the best programming language that can solve the problem at hand, it's important to have an understanding of what programming languages consist of.

Written vs. Spoken languages

As programming languages exist solely in written form, they are classified as written languages. With the ongoing advancements in AI, it might be possible to communicate with computers using natural languages in the future, potentially resulting in programming languages also becoming spoken languages.

Creation and Programming Languages

In the beginning was the Word, and the Word was with God, and the Word was God. He was with God in the beginning. Through him all things were made; without him nothing was made that has been made. (John 1:1-3)

John 1:1-3 says everything was created through God.

God said, Let there be light: and there was light.(Gen 1:3, KJV)

Genesis chapter 1 gives an account of how God created the heavens and earth. It's quite interesting to see how declarative God was when he used his words to create things that didn't exist before. We, as creations of God, see ourselves declaring variables and methods using a programming language.

JC said, Let there be a main method: and there was a main method.

Creation and Intelligent Beings

Although Christian theology and computer science are separate disciplines, they both embrace the notion of creation and intelligent beings. In the context of God's creation, intelligent beings refer to us humans who have the ability to think, talk, and act. Within the domain of computer science, intelligent beings refer to intelligent systems (AI) capable of performing tasks by learning from data, resembling the way humans learn from experience.

Language and Artificial Intelligence

AI will assist developers in automating repetitive and labor-intensive tasks like code debugging and testing, enabling them to focus on the tasks that require one's creativity and insights. AI has become a necessity across diverse realms within computer science such as software engineering, where AI is used to solve problems related to debugging. The connections between Software Engineering and Artificial Intelligence are likely to increase and eventually combine as one field in the future. Therefore, as computer science majors, regardless of our chosen specialization within computer science, the importance of studying AI will grow.

Will programming languages disappear?

Coding is a basic way of communicating with computers, demanding a substantial amount of manual effort. Currently, technologies aimed at reducing labor-intensive coding are in development and are expected to evolve in the future. While it's true that technological advances often replace old methods, it remains uncertain whether programming languages themselves will completely vanish. Nevertheless, one thing certain is that the way in which we interact with computers and write code might undergo changes. We have already witnessed applications using generative AI capable of comprehending high-level program requests and automatically generating the required code. Advancements like AI hold the potential to diminish the need for manual coding, make code writing more user-friendly, and make software development more intuitive.

Possible Exam Questions (at least two)

1. List any similarities between natural language and programming language.

▼ Example answer

- syntax and grammar: Both natural languages and programming languages have syntax and grammar rules that define how words or expressions should be structured to produce correct sentences or code.
- semantics and meaning: In both natural languages and programming languages, the statements and expressions have a certain meaning.

2. What do we learn in PL class and why do we have to take a PL class?

▼ Example answer

In PL class, we learn the principles underlying programming language design and implementation using a combination of interpreters and compilers.

By learning PL,

- we can make the right choice about which programming model and language to use for specific tasks
- we can learn new programming languages quickly

- we can design a light programming language or domain-specific programming languages

참고용