

# Theory of Multi-Paradigm Programming Languages

## Lecture 1-0

## Introduction & Overview

Prof. Moonkun Lee, Ph.D.

Division of Computer Science and Engineering  
Chonbuk National University  
Republic of Korea

# Contents

1. Syllabus

2. Overview

# **1. SYLLABUS**

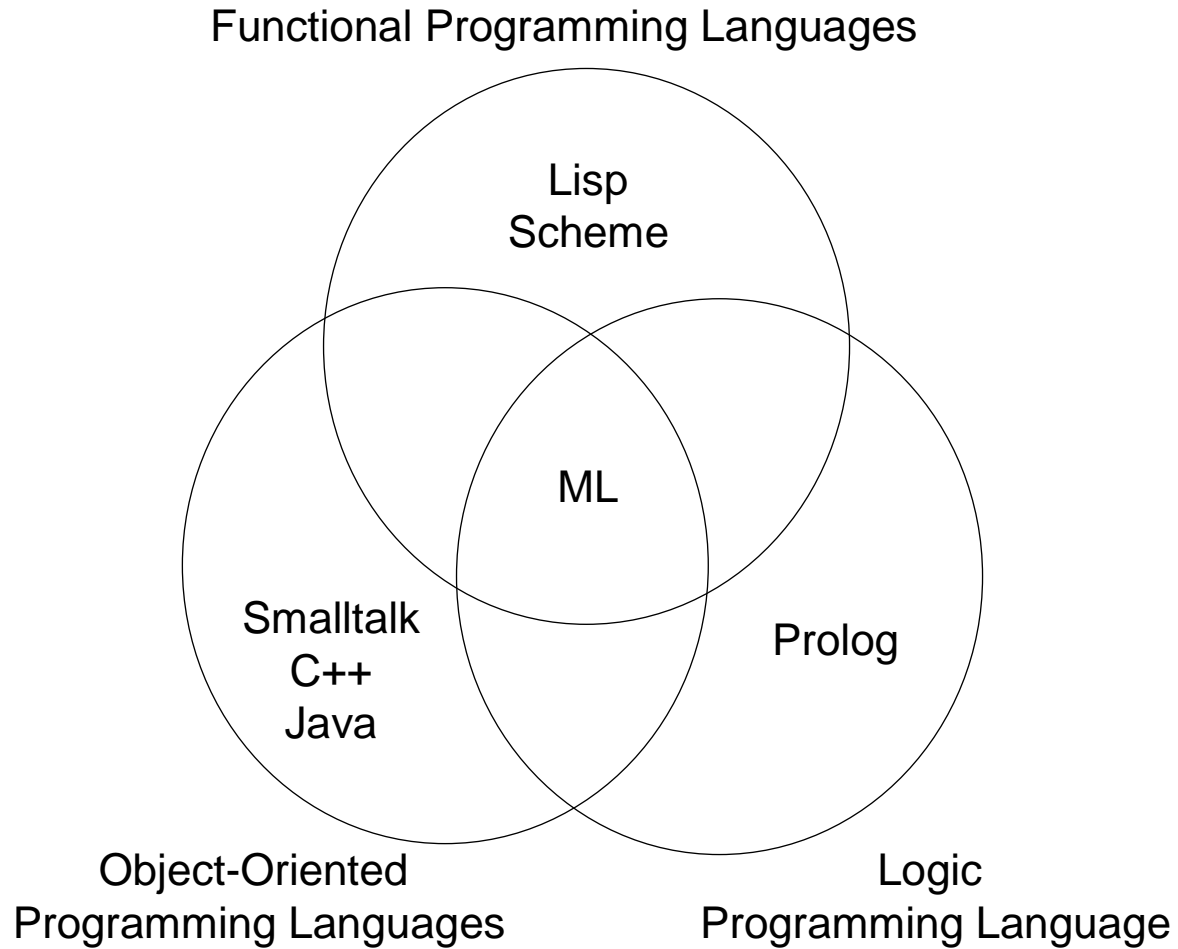
# Instructor

- **Instructor:**
  - Prof. LEE, Moonkun (Ph.D.)
  - Office : College of Engineering, Bldg #7, Rm #624
  - e-mail : moonkun@jbnu.ac.kr
- **Class Room:**
  - College of Engineering, Bldg #7, Rm #302
- **Lecture Hours:**
  - 1) Wed, 12:00pm ~ 12:50pm;
  - 2) Fri, 11:00am ~ 12:50pm
- **Lecture Methods:**
  - 1) Wed [Zoom];
  - 2) Fri [Zoom or Video]
- **Office Hours:**
  - Approx. 30 minutes after class
- **Teaching Assistant:**
  - Junsup Song: junsup@jbnu.ac.kr

# Description

- The purpose of this class is to understand the basic principle of programming languages and practice the principle with the languages on Unix or Linux operating systems.
- The principle includes the definition, history, basic concepts, and the category of the languages, and the practice does the acquisition of the basic knowledge of the languages and constructing codes with the languages on Unix or Linux, as follows:
  - Procedural languages: Pascal, C
  - Functional languages: Lisp, Scheme, ML
  - Object-oriented languages: Smalltalk, C++
  - Logic languages: Prolog
  - Parallel/Distributive languages: CSP, Ada (if time is available ...)
- Phases:
  - 1) The first part of the class: We are learning, with respect to procedural languages, the basic principle of the languages, such as, its design principle, grammar, basic semantics, data types, expression-statements, procedure-environment, etc.
  - 2) The second half of the class: We are learning, with respect to functional languages, abstraction (in the procedure and data perspective), modularity, objects/state, meta-linguistic abstraction, etc.
- **Lab/homework:** We will practice the basic theory of the languages with the over languages on the Unix or Linux environments.

# Domain



# Prerequisite

- This class is one of the most fundamental classes in the area of computer science and engineering, both theoretically and practically. Therefore your hard effort and sincere enthusiasm are definitely required in order to make this course valuable and practical. To make it possible, there are two prerequisite knowledge on the following two subjects: 1) the basic understanding and coding skill of general programming languages, such as, C, and 2) of operating systems, such as Unix or Linux, for coding. If you don't satisfy the prerequisite, you have to talk to me in person to discuss your qualification to take this class.

# Textbooks

- **Recommended Textbooks:**

- 1) 이문근. 『다중패러다임 프로그래밍언어론』. 전북대학교출판원. 2021.
- 2) Kenneth C. Louden. 『Programming Languages: Principles and Practice』, 2nd Edition, Brooks & Cole, 2003.
- 3) Jeffrey D. Ullman. 『Elements of ML Programming』. Prentice Hall, 1997. (교보문고)

- **Reference Books:**

- 4) Ravi Sethi. *Programming Languages: Concepts & Constructs*. Addison Wesley, 1996. (홍릉과학출판사)
- 5) H. Abelson and G. J. Sussman with J. Sussman. *Structure and Interpretation of Computer Programs*. The MIT Press, 1996. (교보문고)
- 6) Samuel N. Kamin. *Programming Languages: An Interpreter-Based Approach*. Addison Wesley, 1990. (교보문고, 홍릉출판사)
- 7) 이문근, 『정형기법』, 전북대학교출판원, 2017.



# Lecture Notes

- The lecture notes for the class will be posted in the lecture page at my homepage: <http://moon.jbnu.ac.kr> .
- Weekly lecture materials will be there at the beginning of each week. You are responsible for your own hardcopy of the notes.

Theory of  
Multi-Paradigm  
Programming Languages

# 다중패러다임 프로그래밍언어론

이문근 저



# Schedule

Weeks	Topics	Homework	Project
1	Overview, Classification		
2	Functional Language : ML (1)		
3	Functional Language : ML (2)		C/C++
4	Functional Language : ML (3)		
5	Functional Language : ML (4)		
6	Functional Language : ML (5)		
7	Functional Language : ML (6)		ML
8	Mid-term Exam		
9	Scheme (1)		
10	Scheme (2)		
11	Object-Oriented Language : Smalltalk (1)		
12	Object-Oriented Language : Smalltalk (2)		
13	Logic languages : Prolog (1)		
14	Logic languages : Prolog (2)		
15	Semantics		
16	Final Exam		

# Grading

	%	Contents	Criteria
Homework	20	Coding works for each languages	
Exam	40	Mid-term (20%): Concepts, definitions, problem solving, coding, etc. Final (20%): Same as the above	Generality
Project	40	Project 1 (20%): C/C++ coding in the middle size Project 2 (20%): ML project in the middle size	Design Documentation Executability

## **2. OVERVIEW**

# Chapters

**제 1장 프로그래밍언어의 원리**

**제 2장 절차중심언어(Procedural Language)**

**제 3장 ML: Multi-Paradigm Programming Language**

**제 4장 Programming Language Paradigms**

# 제1장 프로그래밍언어의 원리

- 1.1 프로그래밍언어의 기본적인 쟁점
- 1.2 프로그래밍언어 정의
- 1.3 프로그래밍언어의 표현 수준(Level of Expression)
- 1.4 프로그래밍언어의 패러다임
- 1.5 결어

# 제 2장 절차중심언어

**2.1 정의(Definition)**

**2.2 동기(Motivation)**

**2.3 속성(Characteristics)**

**2.4 Procedure/Function**



# 제 3장 ML: Multi-Paradigm Lang

3.1 ML언어의 특징

3.2 Basic

3.3 Function

3.4 Input/Output

3.5 More Functions

3.6 사용자 정의 유형 (User Defined Type)

3.7 More Data Structure

3.8 Structure: ML Module System

3.9 결어

# **제4장 Programming Lang Paradigms**

**4.1 Scheme**

**4.2 Prolog**

**4.3 Smalltalk**

# ML Meta-Language Domain

