

# Programming Languages: introduction

국민대학교 컴퓨터공학부

강 승 식

sskang@kookmin.ac.kr

# 프로그래밍 언어 수업은...

- P.L. 관련 주제에 대한 질문(question)
  - *개념(concept)과 키워드(keyword) 중심으로...*
  - *교재가 꼭 필요한가? 모든 지식은 인터넷/웹에 다 있는데...*
- 질문에 대한 답(answer)은 어디서 찾을까
  - 웹에서, Wikipedia에서, etc
- 강의 노트는...
  - *질문을 던지고... 개략적인 답을 간단히...*
  - *자세한 설명과 정답은 스스로 찾도록...*

# P.L. 과목을 배우는 이유

- 생각을 표현하는 능력 향상
- 적합한 언어 선택 능력 향상
- 새로운 언어 학습 능력 향상
- 구현의 중요성 이해
- 이미 알고 있는 언어를 더 잘 사용
- 전반적으로 computing에 대한 이해

# High-level Programming Languages

- **Data abstraction**

- Locations in main memory
  - Data types: char, int, float, double, Boolean, string, pointer
- Give names to a memory block → 변수, 배열, 구조체

- **Process abstraction**

- A sequence of machine instructions
  - High-level expression: numeric, logical
  - Structured constructs: if, for, while
  - Function

- **Data & Process abstraction**

- Class, Module, Package

$x = x + 3$

READ X

ADD X, 3, TMP

STORE TMP, X

```
A_CR = $0D
BSOUT = $FFD2
/
/      LDX #$00
/
LOOP  LDA MSG,X
      BEQ LOOPEND
/
      JSR BSOUT
      INX
      BNE LOOP
/
LOOPEND RTS
/
MSG   .BYT 'Hello, world!',A_CR,$00
```

**Assembler**

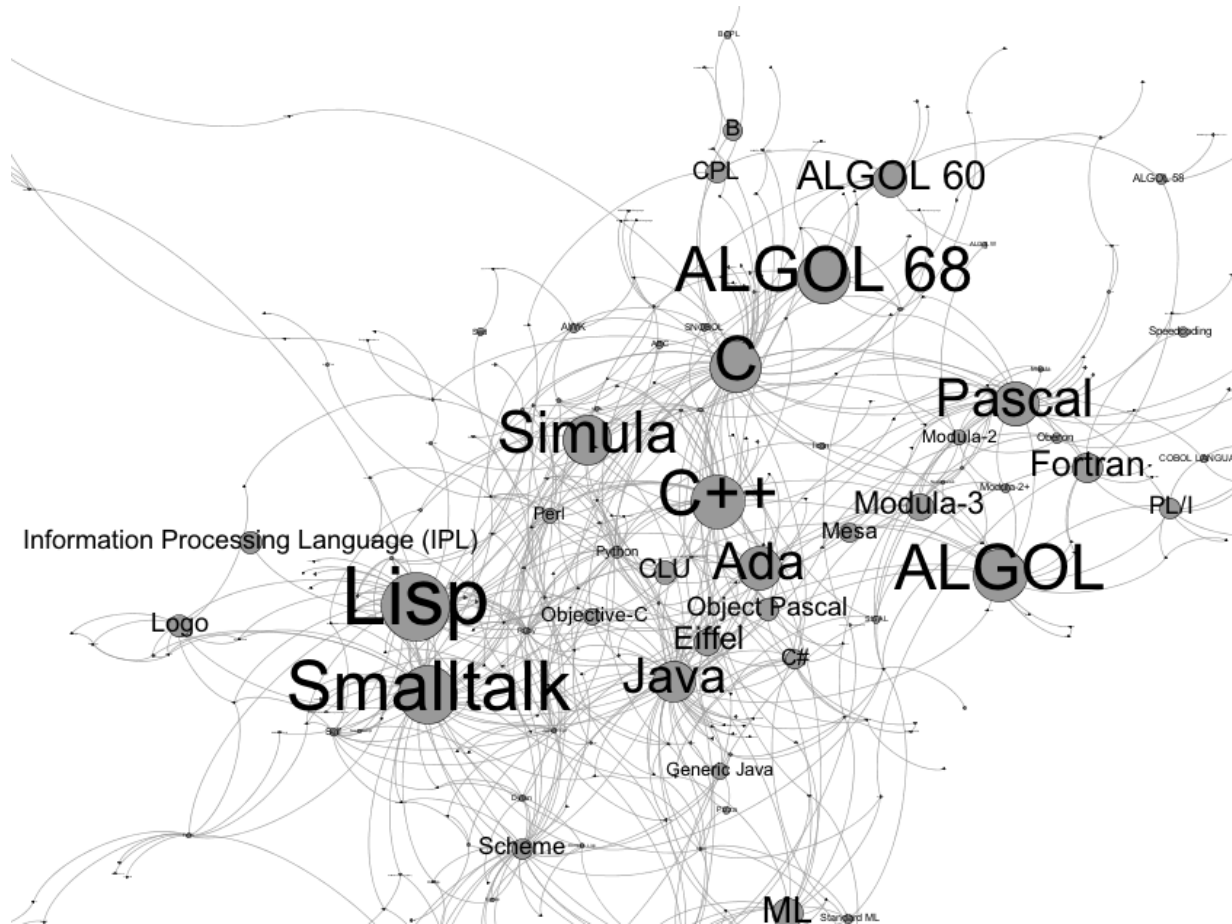
```
begin
  WriteLn('Hello, world!');
end.
```

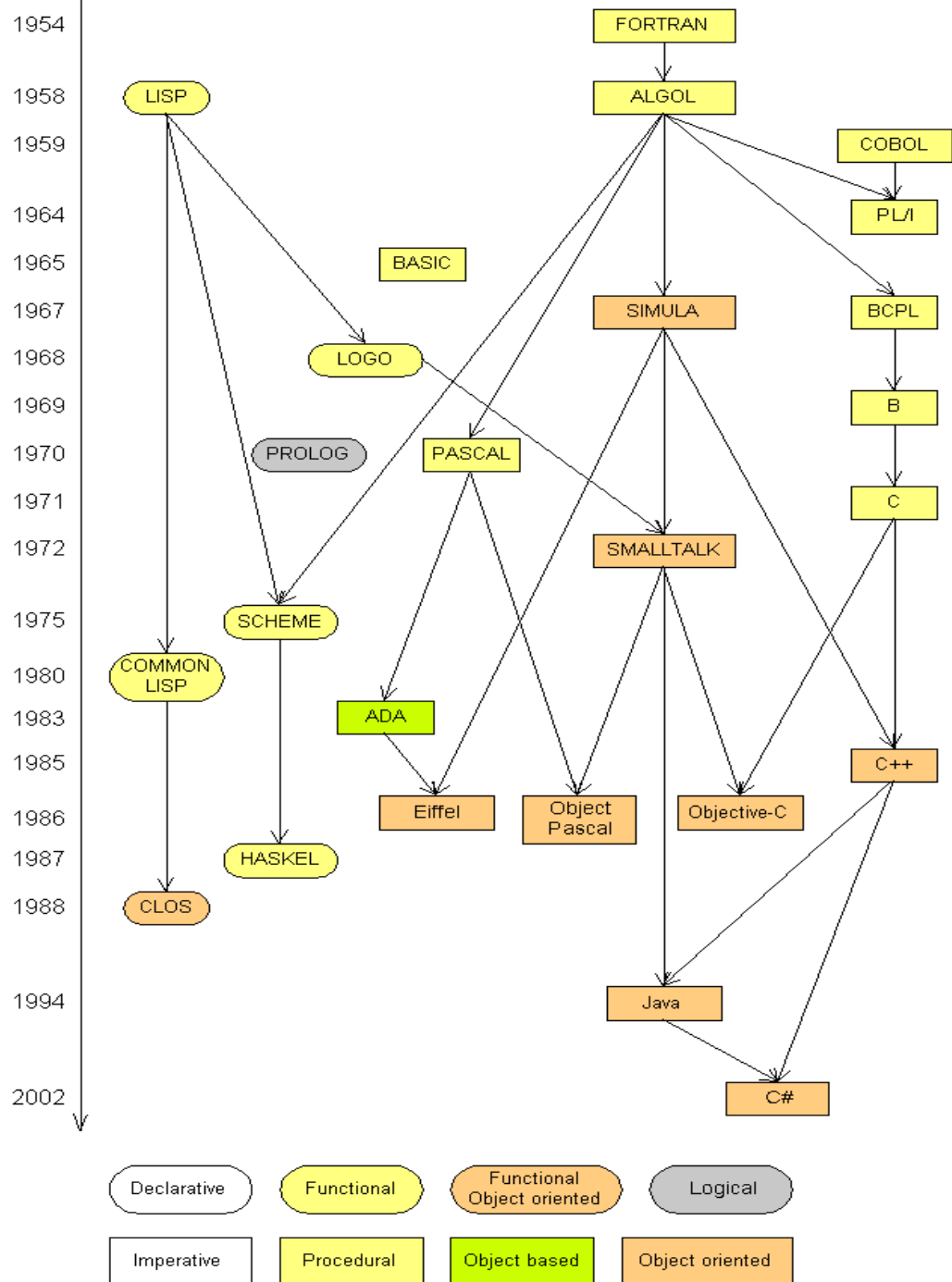
**Pascal**

# Programming Area

- 과학 계산: Fortran
  - 사무 응용: Cobol
  - 인공지능: Lisp, Prolog
  - 시스템 SW: C
- 
- 웹 SW: JavaScript, PHP, ASP
  - 앱 SW: Objective-C, Java
  - 교육용: Basic, Pascal, ML, Haskell
  - Script languages: Perl, Python, Ruby
  - Shell programming: AWK, Tcl/Tk
  - GUI: MFC, C#, Visual Basic

# Programming Languages: Family Tree





# Categories of P.L. -- paradigm

- Imperative(procedural) languages
- Functional languages
- Logic languages
- Object-oriented languages

Imperative/ Algorithmic	Declarative		Object-Oriented
	Functional Programming	Logic Programming	
Algol Cobol PL/1 Ada C Modula-3	Lisp Haskell ML Miranda APL	Prolog	Smalltalk Simula C++ Java



# Programming Language Paradigm

- 컴퓨터 H/W 구조와의 연관성
  - Von Neumann 아키텍처
- S/W 설계 방법론과의 연관성
  - 구조적 설계 vs. 객체지향 설계
- 인간의 사고 방식과의 연관성?



## 존 폰 노이만



수학자

존 폰 노이만은 헝가리 출신 미국인 수학자이다. 무신론자였으나, 나중에 로마 가톨릭 교회 신자가 되었다. 양자 역학, 함수 해석학, 집합론, 위상수학, 컴퓨터 과학, 수치해석, 경제학, 통계학 등 여러 학문 분야에 걸쳐 다양한 업적을 남겼다. 특히 연산자 이론을 양자역학에 접목시켰고, 맨해튼 계획과 프린스턴 고등연구소에 참여하였으며, 게임 이론과 세포 자동차의 개념을 공동 개발한 것으로 잘 알려져 있다. [위키백과](#)

# 언어의 특성

- Overall simplicity(단순성)
- Orthogonality(직교성)
- Expressivity(표현력)
- Portability(이식성)
- Abstraction(추상화)
- Generality(일반성)

# Evaluation of P.L.

- **Readability** 판독력(이해성) 오랫동안 봐도 뭘지 암. Move 100 To A
- **Writability**  작성력
- **Reliability**  신뢰도
- **Cost** 개발비(유지보수비, 실행비용)

# 언어의 구현: Syntax, Semantics

- Compiler implementation
  - Lexical analysis
  - Parsing
  - Intermediate code generation
  - Code generation
    - Symbol table, optimization, error message
- Pure interpretation
  - APL, SNOBOL, LISP
- Hybrid interpretation
  - Perl, Java
  - JIT(Just-in-Time) implementation

Compiled		Interpreted		
C# CLR C++ VB.net	Java	.php	.py	.ruby
CLR	JVM	php	Python	Ruby

# 프로그래밍 환경: 통합 환경

- Editor, Compiler, Linker, Debugger, Profiler 등
  - Microsoft Visual Studio.NET
  - Borland JBuilder
  - NetBeans: Java, JavaScript, Ruby, PHP

1) 프로그래밍 언어의 학습 이유는 무엇인가?

2) 주요 프로그래밍 영역(programming domain)인 과학계산 분야, 사무응용 분야, 인공지능 분야, 시스템 SW 분야, 웹과 앱 SW 분야에서 각각 대표적인 프로그래밍 언어는 무엇일까?

또한, 모든 프로그래밍 영역에 적합한 언어를 설계하여 하나의 언어로 모든 분야의 프로그래밍을 할 경우에 장단점은?

3) 고급언어를 평가하는 주요 기준은 무엇인가?

4) 단순성과 직교성이 언어의 평가기준에 미치는 영향은?

5) 언어를 설계하는데 컴퓨터 구조와 프로그래밍 방법론(구조적 설계, 객체지향 설계 등)이 어떤 영향을 미치는가?

6) 프로그래밍 언어의 평가 기준에서 비용(cost)이란 무엇인가?

7) JIT(Just In-Time) 구현 시스템이란 무엇인가?

8) C 언어에서 직교성(orthogonality) 원칙에 어긋나는 예와 직교성 원칙을 지켰을 때와 그렇지 않을 때 장단점은?

- a. 함수의 return value로 모든 타입을 허용하는 경우
- b. 모든 문장에 대해 return value를 부여하는 경우

9) 타입 검사, 예외 처리, 별칭(aliasing)이 신뢰성(reliability)에 미치는 영향은?

10) 프로그래밍 언어의 평가 기준 4가지(판독성, 작성력, 신뢰성, 비용) 중에서 가장 중요한 것은?

11) 언어의 판독성(readability)은 변수명, 함수명 작성 방법과 문장의 구조(syntax)와 많은 관련이 있다. 명칭을 작성할 때 대소문자를 구분하는 것과 그렇지 않은 것, '\_'를 명칭에 사용하는 것과 그렇지 않을 때 판독성에 어떤 영향을 미치는가?