

CSE 이야기

컴퓨터전공 관심사 (2014-09-12)

- 기업요구 **vs.** 대학교육
 - 프로그래밍 스킬 향상
 - 취업과 대학원
- 전망 있는 분야 (공기업 전산)
- 프로그래머 수명
- 외국 유학 **vs.** 국내 공부
- 학업 이외의 경험 (전공 외의 공부)
- 교우관계
- 국내기업 **vs.** 외국 기업
- 자격증, 공부 **stress** 관리, PPT-교재, 교수면담 예절

SW전공 관심사 (2014-09-12)

- 학교공부와 실무
 - 대학원 진학 (교수희망) 마음가짐
- 자녀 어떤 직업 (유망 직업)
- 외국 진출을 위한 준비
 - 학계 - 외국
- 강사가 하고 싶은 것 (연구)
- 과거 이 시간만은 만족
- 국내 SW 기업체 - game 편중 - 대책

Start Line

나는 누구인가?

- 각자의 잠재적인 천재성

- 천재의 완성을 위하여
 - 무엇을 위해 어떻게 살 것인가?
 - 목표, 동기, 열정

- 우리의 특징
 - 교육환경, 가정

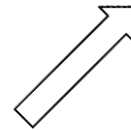
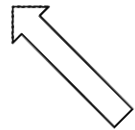
Big Picture

(나는 Computer Scientist!)

CSE: Big Picture

Computer (Machine) + Software

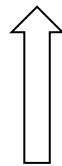
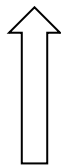
Architecture



Algorithm

컴퓨터 전문지식 기반의

Problem Solving by Programming



Creativity
Curiosity

Logical Reasoning
Domain Knowledge

Expression Skill
Basic Training

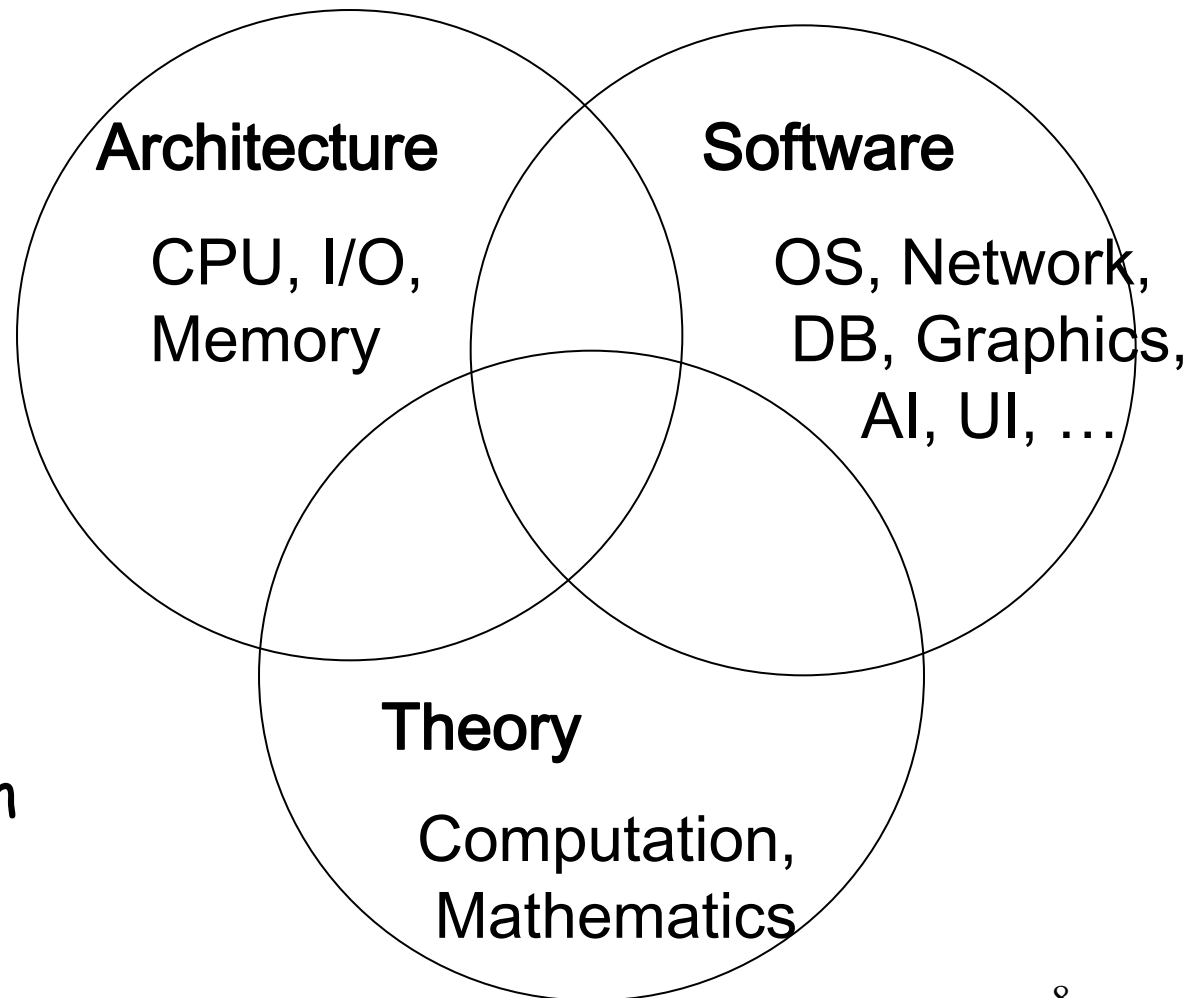
❑ Think like a computer scientist

IT Convergence

□ CSE

□ IT convergence

- Management
- Finance
- Law
- Automotive
- Education
- Transportation
- Silver, ...



What is programmer? (정체성)

❑ What is the problem to solve?

- 인간, 사회, 기술의 연계 (교양)
- 의미 있는 문제들에 대한 끝 없는 탐구

❑ Given a problem, how to solve?

- 논리적 사고 (수학, 과학)
- 컴퓨터 전문지식

❑ On finding a solution, let it happen

- 시인이 시를 토하듯 순식간에 구현 (programming skills)

❖ Extreme form of pragmatics, art, science, intelligence, ..

- What else can be more beautiful?

CSE Success Guide

- ❑ 컴퓨터 전문지식 기반의 소프트웨어/아키텍처 설계
- ❑ 승부의 분기점
 - 학부 1/2 학년: **problem-solving by programming** 훈련
 - 프로그래머가 되는데 전력 다함 (기초이론과 실전)
- ❑ 3/4 학년 (**core** 교과목) - 어떤 문제들을 어떻게 풀었는가?
 - 컴퓨터 전문지식 기반 **problem solving by programming**
- ❑ 석사/박사, 실전: 새로운 컴퓨터 전문지식 창조 (새로운 problem recognition, 새로운 solutions)
 - Creator, most intelligent developer (core library)
- † 10 years of dedication to be first-class programmer
 - 대학원 수준의 교육 중요

CSE: Programmer Line

Game changer
- what to solve

기업 R&D
- SW 유지보수
- SW 신규개발

Academics
- 알고리즘개발

전공교과목
- 이론
- 알고리즘

Tools,
환경,
Libraries

상위 설계
- Req.
- Ext/int spec.

Performance Reliability

Programming: paradigms and concepts
Language: concepts, features, syntax, idioms
Data structures and algorithms

CSE: Course Work (이론강의,기초실습)

Game changer
- what to solve

기업 R&D
- SW 유지보수
- SW 신규개발

Academics
- 알고리즘개발

전공교과목
- 이론
- 알고리즘

Tools,
환경,
Libraries

상위 설계
- Req.
- Ext/int spec.

Performance Reliability

Programming: paradigms and concepts
Language: concepts, features, syntax, idioms
Data structures and algorithms

CSE: Project-Based Training (자기주도)

Game changer
- what to solve

기업 R&D
- SW 유지보수
- SW 신규개발

Academics
- 알고리즘개발

전공교과목
- 이론
- 알고리즘

Tools,
환경,
Libraries

상위 설계
- Req.
- Ext/int spec.

Performance Reliability

Programming: paradigms and concepts
Language: concepts, features, syntax, idioms
Data structures and algorithms

소프트웨어 산업

□ 가장 난이도 높은 두뇌-집약적 산업

- 독점성, 창의성, 서비스/시스템 산업
- 개인적 역량, 사업적 역량, 제도와 시스템

† 두뇌산업 vs. 노동산업

† 국내 수준, 기성세대

- 정상급 전자산업

□ 여러분은 새로운 소프트웨어 세대

- 소프트웨어 전문인력
 - 많은 수련 필요, 충분한 보상
- 불확실성, 제2의 한강의 기적

소프트웨어 공부 어렵다고 느끼는가?

- ❑ 한국과 미국의 차이
- ❑ Successful software company
 - Cash cow: core software system + market + B2B
 - SW 학도: be a part of established system
- ❑ 우리의 교육 및 산업
 - Successful software company
 - Pre-university and in-university curricula
 - Self-driven

컴퓨터공학부 교육 목표

□ 일류 프로그래머 양성?

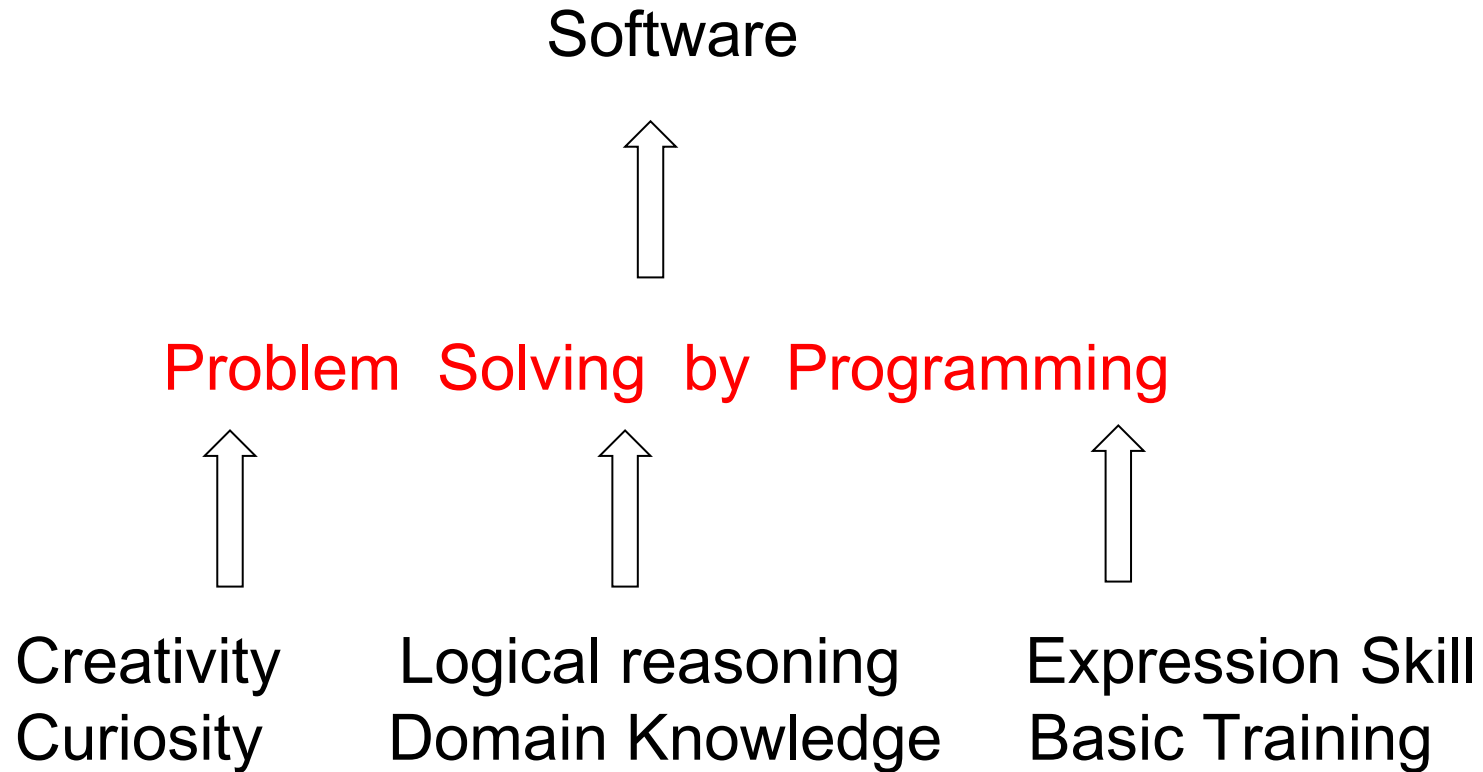
- 컴퓨터전문지식 기반의 **problem solving by programming**

□ 못지 않게 중요한 것은: “큰 교육”

- 나의 인생목표, 적성, 행복을 발견 (한국의 경우 특히 중요)
 - **Mr. Bill Gates** 사례
- 컴퓨터공학부 전공자에게는 무한한 가능성 있음
 - **IT** 정통한 사업가, 투자자, 교육자, 관리자, 경영자
 - **IT** 정통한 산업 디자이너, 법률가, 관료, ...
- 당당한 선택, 도전정신, 탐구하는 마음
- 폭 넓은 교양, 인간과 사회의 이해

□ Problem solving 과 그릇의 크기

비전공자 Programmer



- ❑ Domain-specific problems and ideas

CSE 교과 과정

(컴퓨터전문지식 기반의 problem solving by programming)

Your "C Programming" Class

- ❑ What did you learn?
 - Basic concepts and terminology
 - Basic syntax (statement, function, composite data)
 - Idioms and smallest programs
- ❑ Questions
 - Feel comfortable about C programming?
 - Understand C language?

Your "C Programming" Class

□ 1학년 과목 수강

- C programming 입문을 선언한 상태 (아기가 기는 정도)

□ If didn't study programming further on your won

- Unable to generate meaningful C programs
- Not know C language or high-level programming well

□ 1학년 방학 과제 (전공동아리 필수적)

- Get and read other people's code
- Start writing programs, using existing libraries
 - Then you learn "real" C language and C programming

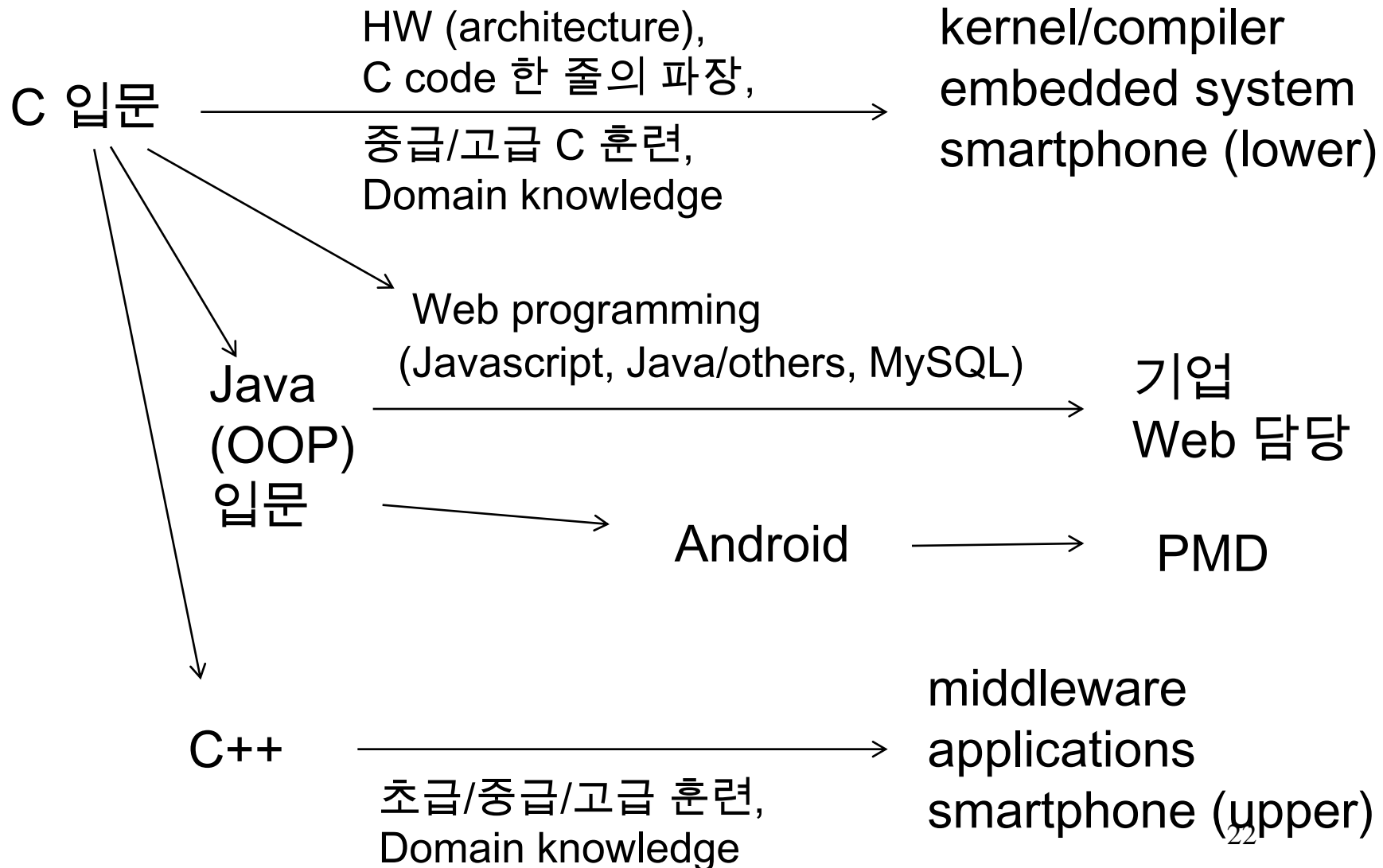
† 고등학교 공부와 대학 공부의 차이

- 자기 탐구, 자기 의지, 자기 관리 필요

어느 중급 C 프로그래밍 책

- Linux 환경, 개발 도구 및 라이브러리 활용 도구 연습
- Free library 이용한 프로그래밍 연습
 - Data manipulation 연습: *GLib* (자료구조론)
 - Database 응용 연습: *mySQL, SQLite* (데이터베이스)
 - Network 프로그래밍 연습: *socket* (컴퓨터네트워크)
 - 과학기술계산 연습: *GSL* (미적분, 선형대수, 수치해석)
 - Image 정보처리 연습: *gdk_pixbuf* (컴퓨터비전)
 - Computer graphic 연습: *Cairo, OpenGL* (컴퓨터그래픽)
 - GUI 프로그래밍 연습: *GTK+, GUI 빌더 (HCI)*
 - Hardware device access
- 아직 전공지식 없어도 라이브러리 사용 시도 가능
 - 1학년 (2학년)때 하지 않으면 프로그래머로 성장 어려움²¹

Simple Roadmap - 3 Popular Languages



3/4 학년 교과목과의 관계

□ Database 사례

- 3/4 학년: 데이터베이스 관련 교과목 수강 + 실전 과제
 - Can use DB and write DB applications
- 석사/박사: 진지한 실전/설계 경험 + 새로운 지식의 창조
 - DBMS (Oracle) 내부 설계, **API** 설계

□ OS 사례

- 프로그래밍 연습하며 **OS API** 사용 시작
- 3/4 학년: OS 관련 교과목 수강 + 실전 과제
 - **OS API** 전문적 사용 (내부 개념 및 동작 이해)
- 석사/박사: 진지한 실전/설계 경험 + 새로운 지식의 창조
 - Kernel internal 및 **API** 설계

CSE 공부의 수준

□ 학사

- 컴퓨터전문지식: Programming 과목, CSE core 과목
- Problem-solving (SW development) 이력
 - 개인 프로젝트 관련 problems, solutions, 결과물 제시
 - 경험/지식/skill set 제시
- 주장: 기업의 문제 해결 가능, 일류 프로그래머 지향

□ 박사 (Ph.D.): 학사 plus

- 전문분야 (architecture, deep learning, 음성인식, DB, ...)
 - 심화된 이론, core library 수정 능력
 - 논문: 신지식 개발 (problem solving) 실적
- 주장: 이 분야 세계 최고 core library 개발 경쟁 가능

□ 석사

교육 과정 vs. 기업 요구

(무엇이 다른가?)

교육과정 vs. 기업요구

- 교과목 모두 **A** 받음 (그 외에는 전공 경험 없음)
 - 이론 (컴퓨터전문지식) 충실
 - 기업이 그를 원할까? 교과과정이 잘못되었나?
- 기업요구: 기업의 상품개발 문제 해결 (**SW** 개발) 가능
 - 실전 중시
- 위 둘의 차이: **problem-solving by programming**
 - **Self-defined software projects** 수행했었나? 몇년간?
 - 이론과 실전의 결합
 - 교과과정 (컴퓨터전문지식) 최소 필요조건
 - 프로그래밍 교과목, **CSE core** 교과목
- 기업의 실무교육은?

Self Projects

- ❑ 자기 주도형 학습 (\leftrightarrow 주입식 교과목)
 - Problem-solving by programming: 이론과 실전의 결합
 - 일류 computer scientist 의 실질적 성장 방법
- ❑ Questions for any CSE major (교과과정 외)
 - 어떤 실전 지식/경험/skill/library set 필요한가?
 - 나의 실전 project roadmap 을 어떻게 찾아 나갈 것인가?
- ❑ "I've never done that"
 - 글 써 본 적 없는 소설가, 연주해 본 적 없는 피아니스트, ..
 - 나는 CSE 에 입문하지 않았음
- ❑ 시작하면 내 삶은 충만하고, 아니면 허무하고 불안함

Self Projects

- ❑ 작은 project 도 몇 달
 - 수업에서 감당할 수 있는 내용이 아님
- ❑ How do we go about this?
 - 실전 프로그래밍 책
 - 동아리, 동호회, 인터넷
 - + 호기심을 기반으로 한 총력전 (남들의 축구에 열광하나?)
- ❑ 컴퓨터공학부의 비전 (실용 학풍)
 - 졸업생들이 (컴퓨터전문지식 기반의) 실전 챔피언
- ❑ 대학 교육의 핵심
 - 해내겠다는 도전 정신
 - 하나씩 정복함에 따라 나는 할 수 있다는 자신감

Programming 배우기 어렵다?

- 서비스 산업, 시스템 산업
 - 전자: 부품 산업
- **Most intelligence-intensive**, 자유도 가장 높은 공학 분야
 - **Project** 관점의 접근 필수
 - 일류 프로그래머 위해 대학원 수준의 교육 중요
- 사회에 진출하는 프로그래머 향한 질문
 - 몇만 줄의 프로그램 읽었나, 만들었나?
 - 어떤 문제를 어떻게 풀었나?
 - 어떤 경험/지식/**skill/library/process set** 갖추었나?
 - **Correct, fast, reliable, elegant software?**

My First Self Project

□ (공부는) 끝없는 질문/탐구

□ **My first project:** 무엇을, 왜, 어떻게?

- 탐구하라! 물어 보라! (사람, 책, 인터넷, 남들의 프로젝트)
- 이것이 가장 중요한 공부 (자기 주도 능력/체제 확립)

† 남들은 무엇을 만들었는가?

- 어떤 문제 어떻게 풀었나, 남들의 코드 읽고 수정해 보았나

† 나는 무엇을 만들고 싶은가?

- 현재의 내가 만들 줄 아는 것은 무엇인가
 - 현재 나의 경험/지식/**skill set** 는
- 그러면 적절한 다음 프로젝트는
 - 이에 필요한 경험/지식/**skill set** 는

Software Success Story (2013-03)

- 17-year old English boy (Nick D'Aloisio)
 - 모바일 앱 섬리 (news summarization application Summly)
 - 2년전 아이폰용 앱 Trimit 출시한 후 벤처투자 받음
 - What problem did he solve?
 - 모바일 기기 대중화 되었으나
 - 기사와 웹페이지는 마우스 클릭에 맞게 제공
 - 태블릿 PC 와 스마트폰에 맞게 짧게 요약
 - Self study of software at age 9
 - First mobile application design at age 12
 - 매각추정대금 약 \$30M (야후 런던법인 근무 예정)
 - 대학에 진학하면 컴퓨터보다는 철학 등 인문학 공부 희망

기업 실무 교육

(무엇이 다른가?)

기업실무교육 vs. CSE 교육

□ CSE 교육

- 컴퓨터전문지식 기반의 **problem-solving by programming**
 - † 연구중심대학의 의미

□ 기업 실무 교육

- Programming

□ 내용이 어떻게 다른가? 추구하는 바가 어떻게 다른가?

공부, 인재

(공부란 무엇인가?)

공부

❑ 공부란 무엇인가?

- 새로운 지식의 탐구 (미지의 세계로의 여행)
- 미래의 현실을 위한 준비
- 끝없는 자기 혁신 (수평, 수직적 성장)

❑ 공부하는 사람으로의 성장 요인

- 호기심, 탐구하는 마음 (자기주도형)
- 주의의 자극/유도/격려/칭찬, *role model*

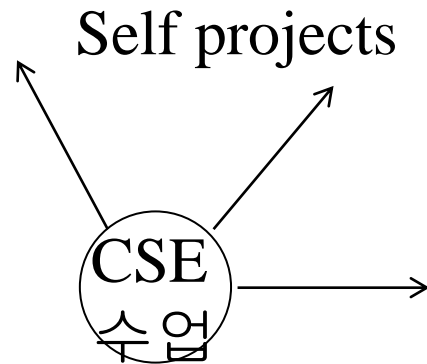
❑ 주입식의 함정

- 주입식으로는 고작해야 3류로 성장
- 호기심 상실, 탐구하는 마음 상실, 자기의 삶 상실

❑ 한국의 현실은?

공부

- 하나를 배우면 열을 안다
- 자극을 받고 뛰쳐나간다
- 학업 이외의 다양한 활동

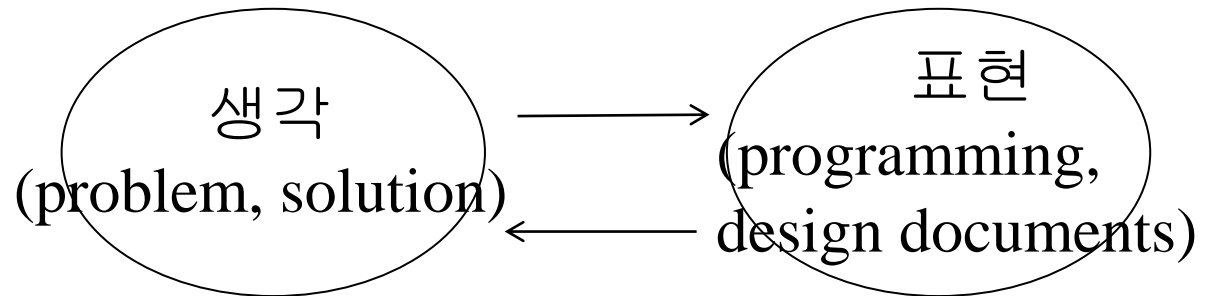


생각과 표현 (자연어, 기계어)

□ 공부: 생각과 표현

- 소설가, 정치가, 연주자, 과학자, ...
- **Computer scientist**

□ Tight loop



□ **Computer scientist** 의 승부 (개인 프로젝트)

- 생각과 표현의 깊이 (**correct, fast, elegant**)
- 크기 (**scale of problem, SW size**)
- 다양성 (**응용분야, 도구, 환경, ...**)
- 결과 **library, 개인 SW 개발 process**³⁷

CSE 공부의 수준

□ 학사

- 컴퓨터전문지식: Programming 과목, CSE core 과목
- Problem-solving (SW development) 이력
 - 개인 프로젝트 관련 problems, solutions, 결과물 제시
 - 경험/지식/skill set 제시
- 주장: 기업의 문제 해결 가능, 일류 프로그래머 지향

□ 박사 (Ph.D.): 학사 plus

- 전문분야 (architecture, deep learning, 음성인식, DB, ...)
 - 심화된 이론, core library 수정 능력
 - 논문: 신지식 개발 (problem solving) 실적
- 주장: 이 분야 세계 최고 core library 개발 경쟁 가능

□ 석사

공부 미흡한 상태에서의 사회 진출

□ 재앙

- 개인 입장에서든, 사회 입장에서든

□ 방법은?

- 졸업을 늦추고, 부족함을 채워라
- 자신을 사랑하고 용서하고 격려하라

전공 외의 공부

(모든 것이 공부다, 평생이 공부다)

컴퓨터공학부 교육 목표

□ 일류 프로그래머 양성?

- 컴퓨터전문지식 기반의 **problem solving by programming**

□ 못지 않게 중요한 것은: “큰 교육”

- 나의 인생목표, 적성, 행복을 발견 (한국의 경우 특히 중요)
 - **Mr. Bill Gates** 사례
- 컴퓨터공학부 전공자에게는 무한한 가능성 있음
 - **IT** 정통한 사업가, 투자자, 교육자, 관리자, 경영자
 - **IT** 정통한 산업 디자이너, 법률가, 관료, ...
- 당당한 선택, 도전정신, 탐구하는 마음
- 폭 넓은 교양, 인간과 사회의 이해

□ Problem solving 과 그릇의 크기

CSE 전공자 커리어 가능성

- ❑ 컴퓨터전문지식 기반의 problem-solving by programming
 - ❑ 컴퓨터전문지식 기반의 problem-solving
 - 새로운 분야의 전문지식 기반의 problem solving
 - ❑ Problem solving
-
- † 그릇의 크기, 변화에 유연함
 - † 일류 대학의 목표는 인재 양성

What are Liberal Arts (교양)?

- ❑ Essential skills for free person
 - To take active part in civic life and public debate
- ❑ Virtuous, knowledgeable, articulate person
- ❑ To form and express well-rounded opinion, via critical thinking

History of Liberal Arts Curriculum

- ❑ Medieval times (5C - 15C): seven liberal arts
 - Trivium: Grammar (Latin), Rhetoric, Logic (verbal arts)
 - Quadrivium: arithmetic, geometry, music, astronomy (numerical arts)
- ❑ Renaissance Humanism in Italy: humanities
 - Grammar, rhetoric, history, Greek, philosophy, poetry
 - Foundation for schooling European elites since 14C
- ❑ Industrial revolution (18 - 19C)
 - Science and Technology Revolution
 - Capitalism

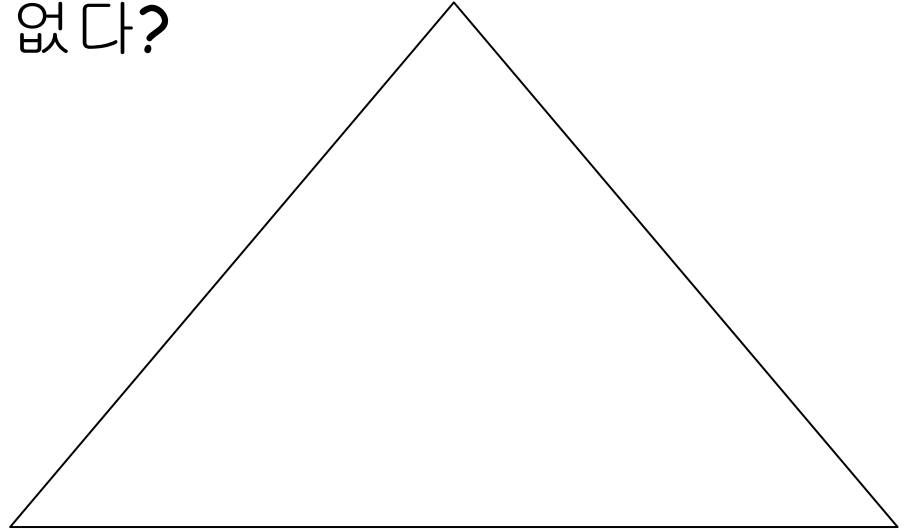
Modern Times Liberal Arts

- ❑ Literature, languages, philosophy, history, mathematics, psychology, science
- ❑ 21 century
 - (말과 글)
 - 사람, 가족, 기업, 정부
 - 기술, 경제, 경영, 자원, 법률, 국제 관계
 - 미디어, 인터넷

기업, 일류, 교우 관계
(어떤 사람에게 끌리는가!)

Pyramid Model: 일류, 삼류

□ 사람이 없다? 일자리가 없다?



□ 무엇이 일류를 만드는가?

- 꿈, 열정, 의지, 노력
- 미친 사람
- 10년 공부

기업

- ❑ 입사 심사 (교우 관계는? 나는 매력적인가?)
 - 학점(컴퓨터전문지식), 실전 능력
 - 열정, 자신감, 도전정신
 - 인간성, 교양
- ❑ 입사 후 오리엔테이션
 - 기본 교육 및 평가 (생각과 표현: 프로그래밍, 말과글)
 - A급 - 핵심 업무에 배치
 - C급 - 행정
 - 이후 격차는 더 벌어짐
- ❑ 인도, 파키스탄에 저가의 1류 프로그래머 많다, 당신은?
 - 입사자 중에 프로그래밍 경험 없는 사람 있다

면접

(어떤 사람을 원하는가?)

내가 면접 심사한다면

❑ 자기 주도형 재능 개발?

- 컴퓨터전문지식 기반의 **problem solving by programming**
- 전공외 공부

❑ 매력적인 동료?

- 성실, 쾌활, 건강, 적극적, 지성

➤ 탐구정신과 자신감 그리고 격조 (상대를 사로 잡는 사람)

면접에 대한 오해

❑ Spec.?

- Only a part of the picture

➤ 내면에서 우러나오는 탐구정신, 자신감, 격조가 중요

❑ 어떤 질문 나올까? 정답을 대야 할 텐데

- 그러나 고급직의 경우, 질문은 **open question**
 - 하나의 정답이 있는 것이 아님
 - 정답보다 반응이 중요: 논리적이고 자신 있는 대응
- 모르면 거꾸로 물어 보라
 - 적절한 질문은 지성인의 상징 (질문을 준비하라)

❑ 자격증? 그것은 통상적으로 기능직의 경우

나는 한양인
(한양의 비전은 무엇인가?)

나는 한양인

- MIT 이야기 나오면, 느낌은?
- 지방대 이야기 나오면?

- MIT, 지방대, HYU 는 과거사 (입학 확정 후 의미 상실)
 - 거기에 과도하게 집착하는 것은?
 - 핵심은 실력
 - 오늘도 나는 누구를 추월하고 누구에게 추월 당한다
 - 실제로는 자신과의 싸움
 - † 일류는 잘 깎여진 보석 (자신에 대한 피나는 세공)

나는 한양인

□ 나의 가능성은?

- 재능 + 노력
 - 여러분 재능 충분
 - 노력하고 있는가?
- 나의 가능성은 무한대

□ 긍정의 메시지: 지금 시작해도 늦지 않는다

- 성장: 초기조건 + 기울기

□ 대기만성

- 묵묵히 하루하루 나의 길을 간다
- 흐르는 물은 선후를 다투지 않는다

나는 한양인

□ 한양대학교 (실용 학풍)

- 컴퓨터공학부 비전

- 나는 컴퓨터전문지식 기반의 실전 챔피언!

전망 있는 분야/직종
(파랑새는 어디에?)

전망 있는 분야/직종

❑ Illusion

- 선택 하나로 평생 편안한 인생 (노력은? 경쟁은? 변화는?)

❑ 타당한 질문인가?

- 점쟁이가 좋아하는 질문

❑ 중요한 질문은?

- 나는 무엇이 하고 싶은가? 나는 무엇을 할 때 행복한가?
 - 대답이 없으면 찾는 것이 최우선
 - 대답을 찾을 때까지 타당성 있는 것 잡아 정진

❑ 자신만의 삶, 도전/모험/탐구 정신

- 유행 따라 남 따라 흘러 다닌다면
- 평균 찾아 다니면 공멸, 나만의 길, 남이 가지 않은 길⁵⁷

인생 101

□ 무엇을 위해 살 것인가?

인생 101

- ❑ 무엇이 1류/성공/행복을 결정하는가?
 - 목표, 사명감, 열정, 체력, 교양, 인간성
- ❑ 목표 없는 사람
 - 오늘은 어디로 갈까?
 - 좋은 인생 목표를 만드는 것이 가장 중요
- ❑ 사명감 (지천명)
 - 개인적 소명, 가정 및 사회에 대한 소명
- ❑ 열정
 - 유행 vs. 내가 하고 싶은 일

❖ 좋은 꿈을 설정하고, 끝없이 도전하고, 성공을 나누라 59

유학 vs. 국내 공부
(공부에도 유학에도 격이 있다)

CSE 공부의 수준

□ 학사

- 컴퓨터전문지식: Programming 과목, CSE core 과목
- Problem-solving (SW development) 이력
 - 개인 프로젝트 관련 problems, solutions, 결과물 제시
 - 경험/지식/skill set 제시
- 주장: 기업의 문제 해결 가능, 일류 프로그래머 지향

□ 박사 (Ph.D.): 학사 plus

- 전문분야 (architecture, deep learning, 음성인식, DB, ...)
 - 심화된 이론, core library 수정 능력
 - 논문: 신지식 개발 (problem solving) 실적
- 주장: 이 분야 세계 최고 core library 개발 경쟁 가능

□ 석사

유학

□ 과거의 whole learning package

- 새로운 문화/언어, 선진 시스템
- 국제적인 human network, 첨단과의 직접 만남
- 전공공부, 졸업 후 현지 활동

□ 유학의 목적이 세계 정상급 편입(즉, 스타 탄생) 이라면

- 사전준비: 프로그래밍 능력, core 교과목, 1류 논문 경험
- 사전준비: problem to solve 준비
- 유학 가서는 정상급으로부터 능력을 확인 받고 동료 편입

□ Internet 시대 (equalizer?)

- 유학과 국내 공부의 차이 점점 줄어듦

❖ “노벨상을 위해 시골학교로 왔다”

대학원 진학 (교수 희망)

- 보다 더 충실한 전공
(컴퓨터 전문지식 기반의 **problem solving by programming**)
 - 새로운 컴퓨터 전문지식 창조
- 보다 더 넓은 기업, 사회, 인생 경험
 - 보다 더 높은 수준의 **problem solving**
 - 보다 더 ...
- 왜 “보다 더”?
 - 많은 예비 스타들에게 직접적인 영향을 주므로

나는 누구인가?
(나는 한국의 대학생)

공부

- ❑ 대학 공부는 고급 인재로 성장하기 위한 공부
 - 자신이 주체가 되어 탐구하는 공부
 - **Elite** 간의 치열한 경쟁
 - **Global** 경쟁
- ❑ 나는 뛰고 있는가? 어느 곳을 향하여?

나는 누구인가

□ 미국

- 개척정신 (frontier spirit)
- “남을 위해 무엇을 할 수 있나?”

□ 일본

- 장인정신 (craftsmanship)
- “남에게 피해를 주지 말라”

□ 한국은?

-

□ 나는 누구인가?

-

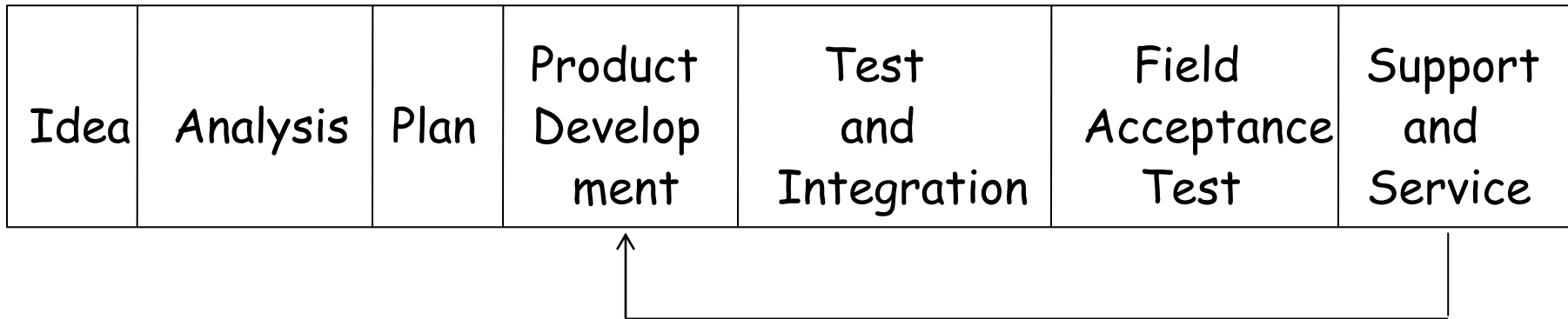
Professional Software Projects

Engineering Design

- ❑ Marketing and requirements analysis
 - New (or existing) product; can we sell it?
 - ❑ Design and implementation
 - External interface
 - How the user will use the product
 - Internal implementation
 - ❑ Testing and release
 - ❑ Continual enhancement
- † Architecture (major interface), architect, abstraction

Software Life Cycle

Decision (Go/No-Go)



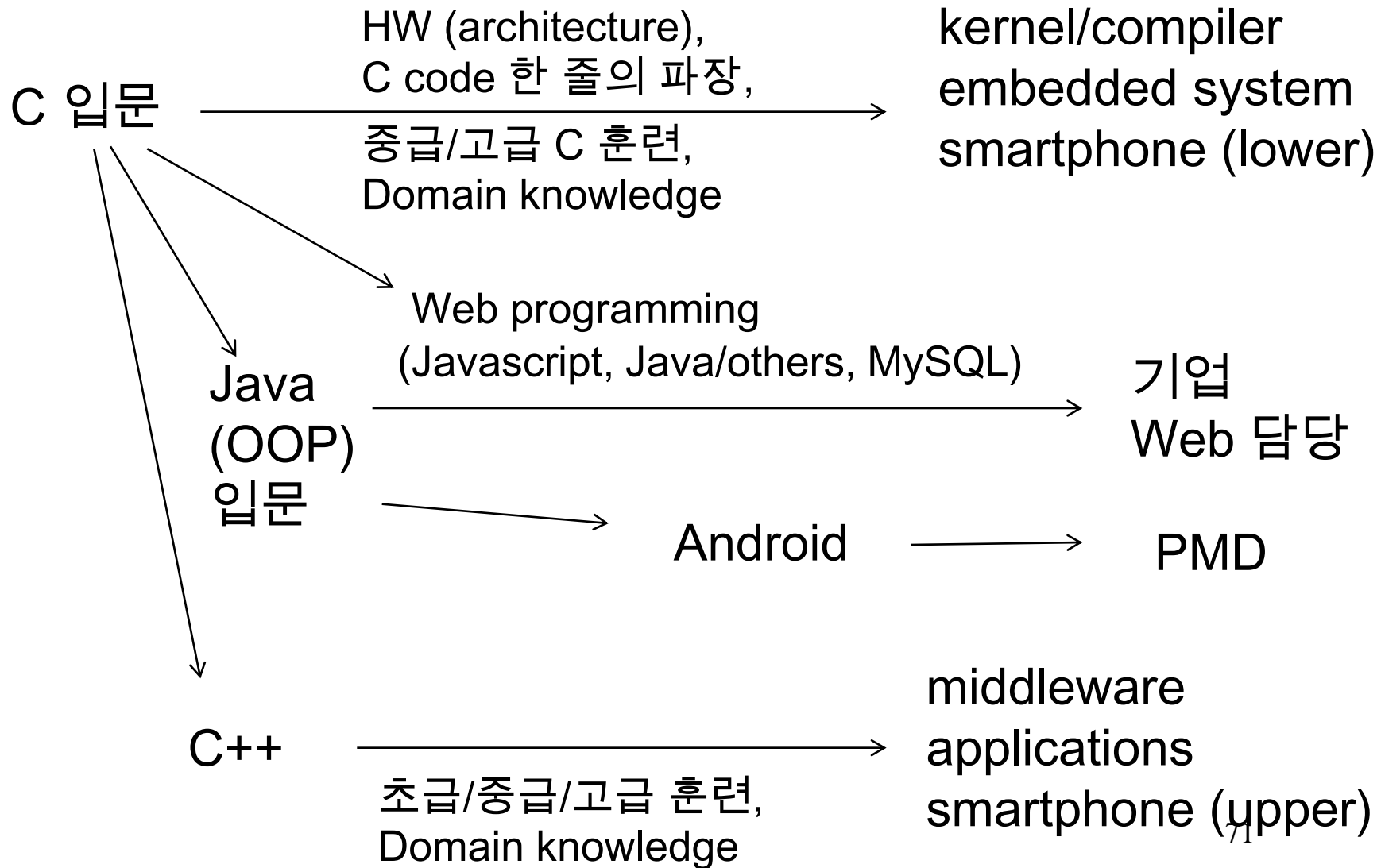
- ❑ What is software engineering?
 - Process (or methodology) for all above activities
 - Procedures, forms, tools, people

Programming Challenge

- ❑ Industry-grade software
 - Correct
 - Fast
 - Reliable
 - Elegant
 - Energy efficiency
 - Solve an important problem
 - Provide interface (GUI, API)

† 컴퓨터 전문지식 기반, 새로운 문제 인식, 해결 및 **SW** 개발

Simple Roadmap - 3 Popular Languages



프로그래머의 수명

프로그래머의 수명

- ❑ 일류 프로그래머 + core software
 - 예: Tizen 성공하고, 나는 Tizen core 개발자
- ❑ 경쟁력 없는 프로그래머
 - 3D 업종

국내 기업 vs. 외국 기업

국내기업 vs. 외국기업

❑ 중요한 질문은?

- 일류 기업인가?
- 내가 원하는 **core** 업무를 맡을 수 있는가?

❑ 외국기업 가기 위한 전제조건

- 해당 언어로 의사 소통 (말과 글)

❑ 참고사항

- 국내의 외국 기업은?
- 해외 기업 입장: “외국인을 불러 쓴다: 어떤 특기 때문에?”
- 해외 유학은 해외 일류 기업 취업의 발판

어느 학부생: “모르는 것이 많아요”

모르는 것이 많아요

□ 막연한 “모르는 것이 많아요”

- 무지, 정체

□ 내가 아는 것이 무엇이고, 모르는 것이 무엇인가?

- 이것을 정교하게 구분할 수 있으면 일류
- 내가 무엇을 공부해야 하는지가 명확함
- 하나를 깨우치게 되면 새로운 궁금한 것이 열 개가 생김
 - “나는 모르는 것이 더 많아요”

□ 무지, 정체의 상태에 있다면

- 아는 것, 모르는 것을 매우 구체적으로 분류해 보라

부록

Feross Aboukhadijeh

- ❑ <http://feross.org/about/>
- ❑ Programmer, designer, teacher, mad scientist
- ❑ 23 years old
- ❑ Stanford alumnus
- ❑ Enjoy working on “mad science”
 - Project that make people say, “Is it possible?”