

# 컴퓨팅 핵심역량 기초 교양교육 과정의 설계

컴퓨팅 핵심역량 교양교육 위원회

서울대학교 기초교육원

2019년 9월

본 보고서는 컴퓨터 소프트웨어와 인공지능 교육을 서울대학교 기초 교양교육으로 어떻게 제공해야 할지를 제안한다. 이러한 교양교육 과정이 제안한 대로 실현된다면 서울대학교 입학생들은 각자의 전공에서 컴퓨터 소프트웨어와 인공지능 기술을 선도적으로 활용할 수 있는 핵심역량을 익히는 체계적인 기회를 가지게 된다.

본 보고서는 2019년 6월부터 8월까지 다섯 번의 위원회 회의를 통해서 정리된 결과를 담았다. 위원회 구성은 다음과 같다:

위원장	이광근	공과대학 컴퓨터공학부
위원	강명주	자연대학 수리과학부
위원	강현구	공과대학 건축학과
위원	강혜정	기초교육원
위원	김수정	미술대학 디자인학부
위원	김성우	공학전문대학원
위원	김용대	자연대학 통계학과
위원	김진공	인문대학 중어중문학과
위원	김태환	공과대학 전기정보공학부
위원	송현오	공과대학 컴퓨터공학부
위원	이병영	공과대학 전기정보공학부
위원	이준환	사회대학 언론정보학과
위원	임용	법학전문대학원 법학과
위원	차주항	자유전공학부
자문	유재준	기초교육원 원장
자문	민은경	기초교육원 부원장

# 차례

1	요약 . . . . .	4
2	위원회 목표/배경/지향점 . . . . .	7
2.1	목표 . . . . .	7
2.2	배경 . . . . .	7
2.3	지향 . . . . .	8
3	신규 교과목 과정/내용/성격/방도 . . . . .	10
3.1	교과 과정: 3 단계 . . . . .	11
3.2	교과 내용: 2 주제 . . . . .	11
3.3	교과 성격: 2 축 . . . . .	12
3.4	교과 방도: 문제중심 . . . . .	12
	3.4.1    문제중심 후보문제 예시 . . . . .	13
	3.4.2    후보문제중심의 교과운영 . . . . .	14
4	신규 교과목 내용 세부 . . . . .	17
4.1	컴퓨팅사고력: 12 개념 . . . . .	17
4.2	데이터사고력: 7 개념 . . . . .	21
5	신규 교과목 목표수준 . . . . .	24
5.1	기초과정 목표치 세부 . . . . .	27
5.2	핵심과정 목표치 세부 . . . . .	28
5.3	고급과정 목표치 세부 . . . . .	29
6	문제중심 교과과정 예시 . . . . .	31
6.1	문제중심 기초버전 예시 . . . . .	31

6.2	문제중심 핵심버전 예시 . . . . .	32
6.3	문제중심 고급버전 예시 . . . . .	34
7	교수/조교/강의자원 수급 . . . . .	36
7.1	수업진행방식 . . . . .	36
7.2	실습 인프라 구성 . . . . .	39
7.2.1	실습 코드 자동 채점기 . . . . .	39
7.2.2	코드 수정돕기 상부상조 플랫폼 . . . . .	41
7.2.3	데이터 사고력 실습 환경 . . . . .	42
7.3	교과목 이름 . . . . .	43
8	부록: 현황 분석 . . . . .	44
8.1	현재 컴퓨터관련 교양교과목 및 이수규정 . . . . .	44
8.2	현재 컴퓨터관련 교양교과목 기초 통계 . . . . .	45
8.3	현재 컴퓨터관련 교양교과목 강의계획서 검토 . . . . .	46
8.4	현재 컴퓨터관련 기타 교과목 및 프로그램 . . . . .	47
8.5	컴퓨터의 개념 및 실습 강의평가 주관식 답변 . . . . .	48

# 1    요약

- 과정: 세 단계 과목을 제공한다.
  - 기초(basic) → 핵심(core) → 고급(advanced)
  - 학생들은 자신에 맞는 단계로 진입하고 필요한 과목만큼 수강한다.
- 내용: 두 주제를 과목의 내용으로 한다.
  - 컴퓨팅 사고력(computational thinking)
    - \* 논리로 표현하는 문제해결 방법
    - \* 논리적으로 컴퓨터 소프트웨어를 만드는 핵심역량
  - 데이터 사고력(data thinking)
    - \* 데이터로 추론하는 문제해결 방법
    - \* 데이터로부터 컴퓨터 소프트웨어를 만드는 핵심역량
- 성격: 두 축을 갖춘 과목 성격을 유지한다.
  - 교육(education)
    - \* 핵심원리와 유효기간이 긴 지식을 전달
  - 교습(training)
    - \* 직접 프로그래밍 실습을 통해 당장 활용가능한 스킬을 전달
- 방도: 문제중심으로 강의를 구성한다.
  - 각 전공에서 흥미롭게 맞닥뜨릴 문제들을 가지고 한 학기를 진행한다(real-world relevance).
  - 그 문제들을 해결해가는 과정에서 자연스럽게 위의 2주제(컴퓨팅 사고력 + 데이터 사고력)와 2축(교육 + 교습)이 전달되도록 구성한다(problem-centered).

- 컴퓨팅 사고력(computational thinking)에서 다루어야 할 개념은 12개다.
  - 데이터 정리하기(data structure), 속내용 감추기/핵심 간추리기(abstraction), 조립식으로 생각하기(composition), 계층쌓기로 생각하기(hierarchy), 끼리끼리 포장하기(module), 반복으로 생각하기(repetition), 재귀적으로 생각하기(induction & recursion), 순서로 생각하기(sequentiality & concurrency), 상태나 값으로 생각하기(state & value), 틀을 짜서 재사용하기(framework), 실행비용 생각하기(cost), 올바른지 확인하기(correctness)
- 데이터 사고력(data thinking)에서 다루어야 할 개념은 7개다.
  - 무작위(randomness), 확률(probability), 두루살펴보기(exploration/sketches), 비교탐색해서 예측하기(prediction by search), 경향파악해서 예측하기(prediction by modeling), 원인해아리기(abduction), 분류하기(classification)
- 강의자원 수급은 다음과 같이 한다.
  - 교수는 강의를 60%, 녹화활용을 40%(flipped learning)정도로 한다.
  - 조교와 학부생 튜터를 활용한다.
  - 실습은 학생들 개개인의 노트북으로 강의실에서 진행한다.
  - 학점체계
    - \* 기초 2:2:0, 핵심 3:3:0, 고급 3:3:0
    - \* 강의시간에 강의 및 실습을 병행한다.
  - 성적부과
    - \* S/U를 기본으로 하고, 학생이 원하면 A-F성적을 준다.
    - \* 또는 전공별로 분반해서 성적을 준다.
- 실습 인프라로 필요한 것은 다음과 같다.

- 실습 코드 자동 채점 시스템: 외주로 개발
    - \* 단, 학생들이 제출하는 정오답 코드들은 서울대가 소유해야함  
(단순 자동채점 시스템을 인공지능화시키는 데 필요)
  - 실습 코드 리뷰 상부상조 플랫폼: [crowdgrader.org](http://crowdgrader.org) 활용
    - \* 학생들이 서로서로 코드를 리뷰하고 코멘트하며 둘째 됨
  - 데이터 사고력(인공지능) 실습을 위한 대형 서버들의 공동활용
- 과목명 후보는 다음과 같다.
  - (참고: “The Beauty and Joy of Computing”) 컴퓨팅의 아름다움  
과 즐거움, 처음 만나는 컴퓨팅, 컴퓨팅의 즐거움, 즐거운 컴퓨팅,  
컴퓨팅 즐기기 등
  - (참고: “Thinking in Computation and Data”) 컴퓨팅 역량키우기,  
컴퓨팅 훈련과 이해, 컴퓨터로 생각하기, 컴퓨팅의 도전 등

## 2 위원회 목표/배경/지향점

### 미션/범위

컴퓨팅 핵심역량 교양교육 틀 디자인

서울대 모든 전공의 학생들이  
컴퓨터라는 도구가 확장시키는 기회를  
선도적으로 활용할 수 있는  
핵심역량을 갖추도록 돕는다

- ▶ 비전공 학부생 대상
- ▶ 틀 디자인 = 과목과정틀 + 과목내용틀 + 과목진행틀
- ▶ Not: 교안 개발 (next project)



### 2.1 목표

컴퓨터 소프트웨어와 인공지능 교육을 서울대학교 기초 교양교육으로 무엇을 어떻게 제공해야 할지를 기획하는 것을 목표로 한다. 이러한 교양교과목을 통해서 서울대 모든 전공의 학생들은 컴퓨터 소프트웨어와 인공지능이라는 도구가 확장시키는 기회를 선도적으로 활용할 수 있는 핵심역량을 갖추게 될 것이다.

### 2.2 배경

소위 “4차산업혁명”이 시작하는 시점에서 서울대가 모든 전공의 학생들에게 컴퓨터 소프트웨어와 인공지능 기술을 쉽게 파악하도록 돋는 과목들을 내실있게 제공할 필요가 있다. 글로벌 선두를 차지하는 인재풀은 각 분야에서 컴퓨터 소프트웨어와 인공지능의 가능성을 독창적으로 연출할 수 있는 인재들이 될

것이기 때문이다. 컴퓨터 소프트웨어와 인공지능이 모든 분야의 성장을 이끄는 중요한 인프라이기 때문에, 각 분야의 미래 인재들은 이 기술들이 가지고 올 다양하고 파급력있는 기회를 적절히 활용할 수 있어야 한다.

### 2.3 지향

본 위원회는 이러한 목표에서 컴퓨팅 핵심역량 교양 과목의 과정/내용/진행에 대한 틀을 디자인하면서, 다음의 지향점들(무순)을 염두에 두었다.

- 지향-1. 미래 선도

다양한 전공의 학생들이 소프트웨어와 인공지능 제작과 이에 필요한 사고방식을 익힐 필요가 있다. 각자의 분야에서 미래에 가능한 응용을 창조하거나 예측할 수 있는 능력을 갖추기 위해서다. 소프트웨어와 인공지능이 모든 분야의 중요한 도구가 되었고 그 분야를 새롭게 이끌 아이디어의 기반이 되고 있기 때문이다.

- 지향-2. 유효기간이 긴 지식

다양한 전공의 학생들에게 소프트웨어와 인공지능 기술의 표면적인 응용뿐 아니라 그 아래를 흐르는 원리를 쉽게 이해시킬 필요가 있다. 지금까지의 정보기술의 성과는 미래에 비하면 미미한 시작이기 때문에, 유효기간이 긴 소프트웨어 지식을 다양한 전공의 학생들에게 전달해 줄 필요가 있다. 학생 각자의 전공 분야에서 오랫동안 유용한 밑거름이 되도록 해야 한다.

- 지향-3. 가능성과 한계의 파악

각 전공의 학생들이 소프트웨어와 인공지능의 가능성뿐 아니라 한계를 이해할 필요가 있다. 소프트웨어와 인공지능의 방식을 적용해서 해결할 수 있는 문제를 적절하게 선별할 수 있는 균형잡힌 안목을 가질 필요가 있기 때문이다. 소프트웨어와 인공지능은 많은 문제를 해결하는 도구이지만 그것만으로 모든 문제를 해결할 수 있는 것은 아니다. 각 분야의

학생들에게 소프트웨어와 인공지능적인 사고방식을 비판적으로 적용하는 균형잡힌 시각을 상기시킬 필요가 있다.

- 지향-4. 디지털 세상의 이해

소프트웨어와 인공지능적인 문제해결 사고방식을 통해서 각 분야의 학생들에게 우리를 둘러싼 디지털 세상을 바라보는 시각을 구체적으로 형성해 줄 필요가 있다. 마치 교양 과학교육이 모든 학생들에게 우리를 둘러싼 자연을 바라보는 시각을 형성해 주는 것과 같다.

- 지향-5. 코딩 실습과 전공현장의 문제 중심

소프트웨어와 인공지능 기술을 이해하고 활용하는데는 학생들이 자신의 전공과 관련된 실제적인 문제를 어떻게 해결할 수 있는지를 구체적인 실습을 통해 익히도록 해야 한다. 실제현장의 문제가 어떻게 해결될 수 있는지를 익히면서 교육의 효과가 극대화 될 것이기 때문이다.

- 지향-6. 컴퓨팅역량 관련 교양과목들의 정비

모든 전공의 학생들에게 “학문의 기초” 트랙의 교양과목으로 정비된 과정을 제공할 필요가 있다. 현재 각 학과나 단과대학별로 제공하고 있는 컴퓨팅역량 기초 과목들은 각 전공에 특화되었다는 장점이 있으나, 컴퓨팅역량의 핵심을 빠뜨림없이 두루 그리고 효과적으로 전달하는 기초 교양과목으로는 아쉬운 면이 있다. 관련 전문가 교수진과 다양한 단과대학 교수진이 함께 팀이 되어 정비된 교과과정을 디자인하는 것이 필요하다.

### 3 신규 교과목 과정/내용/성격/방도

#### 과목 과정/내용/성격/방도 (1/2)

- ▶ 과정: 3 단계
  - ▶ 기초(basic) → 핵심(core) → 고급응용(capstone)
  - ▶ 학생들은 자신에 맞는 단계로 진입
- ▶ 내용: 2 주제
  - ▶ 컴퓨팅 사고력(computational thinking): SW, AI
  - ▶ 데이터 사고력(data thinking): ML, AI



#### 과목 과정/내용/성격/방도 (2/2)

- ▶ 성격: 2 축
  - ▶ 교육(education): 핵심원리, 유효기간 긴 지식
  - ▶ 교습(training): 만들기와 활용하기, 당장 스킬
- ▶ 방도: 문제중심(problem-centered, real-world relevance)
  - ▶ 각 전공에서 흥미롭게 맞닥뜨릴 문제 고안
  - ▶ 그 문제를 해결하면서 2주제 × 2축 익히도록
  - ▶ 단계마다 적절한 “와우” 효과 준비



### 3.1 교과 과정: 3 단계

세 단계 과목으로 과정을 구성한다: 기초(basic), 핵심(core), 고급(advanced).

- 기초(basic) 과목은 컴퓨터나 컴퓨터 소프트웨어의 개발 경험이나 지식이 전혀 없는 학생들이 수강할 수 있도록 기초적인 소양을 전달하는 데 집중한다.
- 핵심(core) 과목은 어느 정도 컴퓨터와 컴퓨터 소프트웨어의 개발 경험이나 지식이 조금 있는 학생들에게 컴퓨팅 핵심역량을 균형있게 전달하는 데 집중한다.
- 고급(advanced) 과목은 컴퓨터와 컴퓨터 소프트웨어의 개발 경험이나 지식을 위의 핵심과목 수준으로 갖춘 학생들에게 빅데이터나 인공지능 소프트웨어 개발 및 응용 역량을 전달하는 데 집중한다.

### 3.2 교과 내용: 2 주제

두 개의 주제로 내용을 구성한다: 컴퓨팅 사고력(computational thinking), 데이타 사고력(data thinking).

- 컴퓨팅 사고력(computational thinking): 컴퓨터라는 도구에게 어떤 일을 시키고자 할 때, 그 일을 어떻게 하면 되는지를 논리적으로 궁리하고 그렇게 궁리한 결과를 컴퓨터 소프트웨어로 표현할 때 동원하는 생각방식에 대한 것이다.
  - 논리로 표현하는 문제해결 방법: 일을 수행하는 방법을 논리적으로 표현하는 전통적인 소프트웨어 제작에 동원되는 생각방식이다. 논리적으로 컴퓨터 소프트웨어를 만드는 핵심역량에 대한 내용이다.
- 데이타 사고력(data thinking): 컴퓨터라는 도구에게 어떤 일을 시키고자 할 때, 그 일을 어떻게 하면 되는지는 알 수 없지만 주어진 많은 데이터로 부터 유추과정을 통해서 컴퓨터 소프트웨어를 구현할 때 동원하는 생각방식에 대한 것이다.

- 데이터로 추론하는 문제해결 방법: 일을 수행하는 방법을 논리적으로 표현할 수는 없지만, 빅데이터(big data) 기술이나 기계학습(machine learning) 기술 -소위 “서당개 5만년” 컴퓨터 훈련 기술- 을 이용해서 많은 정답 데이터로부터 원하는 일을 해내는 인공지능 소프트웨어를 만드는 핵심역량에 대한 내용이다.

### 3.3 교과 성격: 2 축

교육(education)과 교습(training)의 성격을 모두 갖춘 교과과정이 되도록 한다.

- 교육(education): 다양한 전공의 학생들에게 소프트웨어와 인공지능 기술의 표면적인 응용뿐 아니라 그 아래를 흐르는 원리를 쉽게 이해시킬 필요가 있다. 지금까지의 정보기술의 성과는 미래에 비하면 미미한 시작이기 때문에, 유효기간이 긴 소프트웨어 지식을 다양한 전공의 학생들에게 전달해 줄 필요가 있다. 학생 각자의 전공 분야에서 오랫동안 유용한 밑거름이 되도록 해야 한다.
- 교습(training): 소프트웨어와 인공지능 기술을 이해하고 활용하는데 당장 필요한 실습 기술을 전달한다. 실제 소프트웨어를 만들어보거나 있는 것을 활용해 보는 구체적인 실습을 통해서, 두려움을 극복하고 당장 써먹을 수 있는 기술들을 익히도록 한다. 특히, 학생들이 자신의 전공과 관련된 실제적인 문제를 어떻게 해결할 수 있는지를 구체적인 실습을 통해 익히도록 한다.

### 3.4 교과 방도: 문제중심

각 전공에서 흥미롭게 맞닥뜨릴 문제들을 가지고 한 학기를 진행한다. 그 문제들을 해결해가는 과정에서 자연스럽게 위의 과목내용 2주제(컴퓨팅 사고력 + 데이터 사고력)와 과목 성격 2축(교육 + 교습)이 전달되도록 구성한다

(problem-centered). 그 문제를 해결하는데 풀어야 할 하부 문제들이 여럿 출현하게 될 것이고 이런 다양한 부품문제들을 해결하면서 위의 과목내용 2주제와 과목성격 2축이 효과적으로 전달될 것이다.

소프트웨어와 인공지능 기술을 이해하고 활용하는데는 학생들이 자신의 전공과 관련된 실제적인 문제를 어떻게 해결할 수 있는지를 구체적인 실습을 통해 익히게 된다(real-world relevance). 실제현장의 문제가 어떻게 해결될 수 있는지를 익히면서 교육의 효과가 극대화 될 것이다.

### 3.4.1 문제중심 후보문제 예시

다음의 16개 문제는 위원회에서 각 전공 학생들이 관심을 가질 수 있는 문제들을 예를 들어 고안해 본 것이다. 한 학기 과정 내내 이러한 문제를 해결해 가는 과정이 중심이 되어, 과목에서 목표로하는 컴퓨팅 사고력과 데이터 사고력을 학생들에게 전달하게 된다. 학생들은 각 전공과 관련될 듯한 문제이면서 주변에서 겪는 문제를 컴퓨팅 사고력과 데이터 사고력을 동원해서 해결해가는 과정을 맟는다.

#### 문제중심: 후보문제들 (1/2)

1. 시/그림의 작가를 유추하자 (진품명품 감정인을 만들자)
2. 로미오와 줄리엣의 비극이 없는 짹짓기를 해주자
3. 나의 카톡 기록을 보고 주소록 친구들을 자동 분류하자
4. 창비시집에 출현하는 모든 단어들의 빈도 변화를 시간대로 알아보자
5. 다툼에서 양방이 사실로 인정하는 것을 찾는 게임을 만들자
6. 국회의원/공무원 한 명을 찾아 컴퓨터로 대체하자
7. 카톡에 가위바위보 서비스를 만들자
8. 조선시대 과거급제자 친인척관계를 파악하자
9. 구한말 흑백사진을 총천연색 자연스런 칼라사진으로 변환하자

## 문제중심: 후보문제들 (2/2)

10. 조선왕조실록에서 분노 많은 왕들을 순서대로 랭킹하자
  11. 고려사에서 대신들과 회의가 가장 길었던 왕들을 순서대로 랭킹하자
  12. 서울대입구에서 등교 카풀을 유도하는 앱/게임을 만들자
  13. 서울대학교 전공별 수강규칙 문답을 해주는 챗봇을 만들자
  14. 매일 다니는 길을 벗어난 흥미로운 새 경로를 추천하는 앱을 만들자
  15. 댓글 폰트를 화장실낙서 수준으로 보여주는 화장술을 고안하자
  16. 창업 아이템으로 발전시켜라 (해커톤 개최)

### 3.4.2 후보문제중심의 교과운영

예를 들어, 위 문제들의 해결과정이 어떻게 학생들로 하여금 컴퓨팅 사고력과 데이터 사고력을 익히게 하는지를 개관하면 다음과 같다.

- 문제 “시/그림의 작가를 유추하자 (진품명품 감정인을 만들자)”

주어진 시를 입력으로 주면 어느 시인의 작품인지를 판단하는 소프트웨어를 만들어 본다. 기본적인 컴퓨팅 사고력도 익히게 되지만 우리나라 시들을 학습 데이터로 해서 시와 시인의 연관관계를 자동으로 파악하는 기계학습 기술을 활용하는 경험을 하게 된다.

- 문제 “로미오와 줄리엣의 비극이 없는 짹짓기를 해주자”

소개팅을 받는 두 그룹으로 부터 상대방 그룹에 대한 선호도 순서를 입력으로 받아서, 선호도를 반영한 짹짓기를 해 주는 소프트웨어를 만들어 본다. 짹짓기의 조건으로 모두가 로미오와 줄리엣의 비극은 없어야 한다. 어느 남녀 한 쌍도 둘 다 서로보다 덜 선호하는 사람이 짹이 되는 불행이 있어서는 안된다. 컴퓨팅 사고력의 대부분을 익힐 수 있는 문제이다.

- 문제 “나의 카톡 기록을 보고 주소록 친구들을 자동 분류하자”

학생의 카톡에 지금까지 주고받았던 문자들을 살펴서 친구들을 자동으로 분류해주는(격이없는 친구, 어색한 친구, 공식적인 관계, 애인, 썸타는 친구 등) 소프트웨어를 만들어 본다. 기본적인 컴퓨팅 사고력도 익히게 되며 데이터 사고력중 기계학습과 분류하기 기술을 활용하는 경험을하게 된다.

- 문제 “다툼에서 양방이 사실로 인정하는 것을 찾는 게임을 만들자”

두 사람 사이에 분쟁이 생겼을 때, 양측이 제출한 관련자료를 보고 서로가 사실로 인정하는 항목을 자동으로 찾아주는 소프트웨어를 만들어 본다. 제출한 자료는 모두 텍스트 자료라고 한정하고, 데이터 사고력 중 기계학습을 이용한 자연어처리 기술과 컴퓨팅 사고력의 많은 것을 경험하게 된다.

- 문제 “카톡에 가위바위보 서비스를 만들자”

가위바위보는 두 사람이 동시에 같은 장소에 있어야 공정함을 확인할 수 있는 게임이다. 이 게임을 카톡에서도 할 수 있게 하는 소프트웨어를 만들어 본다. 중간 심판을 두지 않고 시공간 차이를 극복해서 모두가 믿을 수 있는 공정한 가위바위보를 가능하게하는 소프트웨어를 만드는 것이다. 학생들은 컴퓨터를 이용해서 논리적인 방식으로 시공간을 극복하는 예를 겪게 되고, 컴퓨팅 사고력의 대부분을 경험하게 된다.

- 문제 “조선시대 과거 급제자 친인척관계를 파악하자”

조선시대 과거 급제자의 친인척관계를 파악하는 소프트웨어를 만들어 본다. 조선시대 과거 급제자의 데이터와 족보 데이터를 모두 처리하는 과정을 통해서 학생들이 데이터 사고력과 컴퓨팅 사고력의 대부분을 익히게 된다.

- 문제 “조선왕조실록에서 분노 많은 왕들을 순서대로 랭킹하자”

한자어로 화를 표현하는 단어나 문구들이 조선왕조실록에서 왕의 말로 나타나는 경우를 모두 찾아 그 빈도에 따라 왕을 차례대로 순서짓는 소프트웨어를 만들어 본다. 학생들은 컴퓨팅 사고력의 대부분을 익히게 된다. 데이터 사고력에서는 화를 내는 사람이 임금인지를 확인하는 자연어처리 과정을 동원하게 된다.

- 문제 “서울대입구에서 등교 카풀을 유도하는 앱/게임을 만들자”

매 학기 서울대입구역 버스정거장은 등교하려는 학생들로 늘 병목현상을 겪는다. 아침마다 학생들은 긴 줄을 서서 버스를 기다리고 있는 반면 그 앞 도로에는 한 명만 탄 자가용들이 줄줄이 서울대로 향하는 어이없는 상황을 소프트웨어와 인공지능기술로 개선해 본다. 등교하는 학생들과 서울대행 자가용을 짹지어주는 앱을 만들어보면서 학생들은 대부분의 컴퓨팅 사고력과 데이터 사고력을 익히게 된다.

- 문제 “서울대학교 전공별 수강규칙 문답을 해주는 챗봇을 만들자”

서울대 전공별로 다양한 수강규칙이 존재한다. 자연어로 주고받으며 학생들의 수강규칙 문의에 답하는 소프트웨어를 만들어 본다. 학과의 졸업 규정을 파악하고, 다양한 문의를 이해하고 논리적으로 답을 찾는 과정이 필요하다. 학생들은 컴퓨팅 사고력의 대부분과 데이터 사고력에서 자연어처리 기술을 경험하게 된다.

- 문제 “매일 다니는 길을 벗어난 흥미로운 새 경로를 추천하는 앱을 만들자”

학생들의 핸드폰에 남은 움직임 데이터를 받아 움직임 경로 주변의 흥미로운 장소를 들리도록 새로운 경로를 추천하는 소프트웨어를 만들어본다. 학생들은 핸드폰의 자료를 활용하는 방법과 상용 지도에 있는 정보를 참고하여 문제를 해결하게 된다. 학생들은 대부분의 컴퓨팅 사고력을 익히게 된다.

## 4 신규 교과목 내용 세부

## 4.1 컴퓨팅사고력: 12 개념

컴퓨팅사고력: 부품 12개	
참고: [L444.200 컴퓨터과학적 사고와 실습] 2017	
c1	데이터 정리하기
c2	속내용 간추기/핵심 간추리기
c3	조립식으로 생각하기
c4	계층쌓기로 생각하기
c5	끼리끼리 포장하기
c6	반복으로 생각하기
c7	재귀적으로 생각하기
c8	순서로 생각하기
c9	상태나 값으로 생각하기
c10	틀을 짜서 재사용하기
c11	실행비용 생각하기
c12	올바른지 확인하기

- 데이터 정리하기(data structure)
    - 자료를 정리하는 지혜다. 정리한 모양새(구조)에 따라 일을 편하게 할 수 있게된다. 하려는 일에 따라 가장 적절한 자료 정리의 모양새가 있다.
    - 비유: 책방에 책이 아무렇게나 꽂혀있다면 책을 찾기 힘들다. 식당 줄이 일렬이 아니라면 혼돈이 온다.
  - 속내용 감추기/핵심 간추리기(abstraction)
    - 복잡한것을 다루는 지혜다. 외부에는 속구현을 알려주지말라. 속구현이 어떻게 되었는지 몰라도 만든것을 사용할 수 있도록 한다.
    - 비유: 자동차운전은 자동차 엔진이 어떻게 만들어졌는지, 브레이크가 어떻게 만들어졌는지 몰라도 사용할 수 있다.

- 조립식으로 생각하기(composition)

- 독립적인 부품들을 조립해서 전체를 만드는 지혜다. 전체가 부품들로 분해될 수 있고 부품들을 제 역할에 맞게 조립하면 원하는 전체가 된다. 전체 방법은 부분 방법들의 조립이다.
- 비유: 수능시험을 잘 보는 방법은 각 과목시험을 잘 보는 것이다. 자동차는 엔진, 바퀴, 샤시, 킥질, 의자 등으로 조립된다. 대학은 단과대학들과 행정지원 조직으로 조립된다.

- 계층쌓기로 생각하기(hierarchy)

- 조립하는 과정을 여러 계층으로 쌓아가는 지혜다. 만든것을 부품으로 활용해서 더 큰 것을 만들고, 이렇게 만든 것이 또 부품이되어서 더 큰 것을 만든다. 1층에서 만들어서 2층으로 올리고, 2층에서는 다시 반복하면서 차곡차곡 올라간다. 맨 윗층의 것이 우리가 원하는 것이다. 위층에서는 아래 층의 속구현을 몰라도 쌓아올라갈 수 있다.
- 비유: 신경, 뇌, 뼈, 근육, 소화기관, 피부가 사람을 만들고, 사람들이 마을을, 마을들이 나라를, 나라들이 문명을 만든다. 각 계층의 속내용을 감추며 위층으로 위층으로 더 복잡한 것을 만들어간다.

- 끼리끼리 포장하기(module)

- 비슷한 것끼리 모아서 포장해 놓고 정리해 놓는 지혜다. 많이 모여 있는 것들을 잘 정리하려면, 유사한 것을 한데 모으고 박스에 넣고 박스에 표시를 해둔다.
- 비유: 이삿짐 포장할 때 “거실”, “부엌” 등으로 박스를 마련해 놓고 포장하고 이사하고 짐을 풀듯이.

- 반복으로 생각하기(repetition)

- 같은 작업을 반복하면서 일을 마치는 지혜다. 같은 일을 다른 데이터를 가지고 반복하면서 일을 끝마칠 수 있다. 아는 것만 알고 갔던 곳만 가면서 일을 끝마친다.
- 비유: 밥을 먹는 일은 수저와 젓가락으로 밥과 반찬을 입에 넣은 후 씹어삼키는 작업의 반복이다.

- 귀납으로 생각하기(induction & recursion)

- 데이터가 귀납적으로 만들어지는 종류들이 있다. 리스트는 리스트로 만든다. 트리는 트리로 만든다. A종류의 데이터를 생각하자. A 데이터를 만드는 기초적인 부품이 있고, 다음부터는 그렇게 만든 A 데이터로 새로운 A 데이터를 만든다.
- 비유: 나무 가지를 보자. 왼쪽 오른쪽 나무가지들로 나무가지가 만들어 져 있다. 양파를 보자. 양파를 한 껌풀 벗기면 다시 양파다. 양파로 양파를 만든 것이다.

- 순서로 생각하기(sequentiality & concurrency)

- 일의 순서를 생각하는 지혜다. 일을 쪼개고 쪼개진 일의 순서를 생각한다. 대부분의 일에는 순서가 있다. 어떤 일을 마쳤을 때에만 그 다음 일을 할 수 있다. 이전 일의 결과가 다음 일에 꼭 필요하기 때문이다.

일의 순서가 구지 없다면 한꺼번에 진행해도 된다. 쪼개진 일들의 순서가 필요 없는 경우가 있다. 이전 일의 결과가 다음 일에 필요한 것이 아니라면 순서없이 동시에 진행해도 된다.

- 비유: 비빔국수를 요리하는데 순서대로 할 일이 있고 동시에 해도 되는 일이 있다. 물을 끓인 후 국수를 넣고 삶아야 한다. 한편, 밀가루 반죽과 소스를 만드는 일은 둘이 동시에 진행해도 된다.

- 상태나 값으로 생각하기(state & value)

- 변하는 상태를 생각하거나 변함없는 값을 생각하며 일하는 지혜다. 물건에 변화를 일으키면서 일할 수도 있고, 값을 만들고 만든 값으로 새로운 값을 계산하는 과정으로 일할 수도 있다.
- 비유: 장바구니라는 물건은 장을 본 물건들로 채워지며 상태가 변한다. 그러면서 장보기를 마친다.  
변하지 않는 값으로 생각하는 예로, 각 부서가 업무 매뉴얼을 만드는 과정을 보자. 옆 부서에서 만든 매뉴얼을 가져다가 우리 부서에 맞는 매뉴얼을 만들지만 옆 부서의 매뉴얼을 바꾸지는 않는다.

- 틀을 짜서 재사용하기(framework)

- 자주 쓰는 것을 재사용하기 쉽게 준비해놓는 지혜다. 지금 만든 것을 지금의 경우에만 쓰게하지 말고, 지금의 상황과는 동떨어뜨려서 일반적인 경우로 확장해서 준비해놓는다. 재사용할 때마다 사용되는 상황이 다를 것이다. 다른 점을 반영해서 경우마다 조금씩 다른 일을 할 수 있도록 준비해 놓는다.
- 비유: 떡살을 생각하자. 매번 같은 모양의 떡을 만들지만 넣는 재료에 따라 다른 떡을 찍어줄 것이다.  
공교육 시스템 같은 것이다. 유아교육, 초등교육, 중고등교육의 틀이 있다. 각 교육과정에 새로운 비중의 과목과 내용이 채워지면서 새로운 시대에 맞는 교육내용을 실현한다.

- 실행비용 생각하기(cost)

- 컴퓨터가 소프트웨어를 실행하면서 시간과 메모리를 얼마나 쓸지 어림잡는 지혜다. 만든 소프트웨어가 실행되면 얼마나 시간과 메모리를 소모할까. 입력의 크기에 따라 비용이 어떻게 될지를 생각하면서 더 나은 방법을 찾는 길잡이다.
- 비유: 1부터 100까지 그녀가 마음속에 가지고 있는 숫자를 알아맞추는 방법을 생각하자. 1부터 차례로 물어가는 방법도 있고, 매번 그 포위망을 반으로 좁혀가는 방법도 있다.

- 올바른지 확인하기(correctness)

- 우리가 생각한 바를 틀림없이 소프트웨어로 표현했는지 확인하는 지혜다. 컴퓨터는 소프트웨어 그대로를 맹목적으로 실행할 뿐이다. 우리가 짠 소프트웨어에 실수가 있다면 컴퓨터는 그 실수를 고스란히 드러낸다.
- 비유: 로켓(소프트웨어)을 잘못만들면 자연(컴퓨터)이 실행시키는 그 로켓은 상승중에 폭발하는 것과 같은 이치다.

## 4.2 데이터사고력: 7 개념

### 데이터사고력: 부품 7개

참고: *Computational and Inferential Thinking*, textbook for UC Berkeley / 김용대교수, 송현오교수, 이재용교수

d1	무작위	randomness
d2	확률	probability
d3	두루살펴보기	exploration/sketches
d4	비교탐색해서 예측하기	prediction by search
d5	경향파악해서 예측하기	prediction by modeling
d6	원인해아리기	abduction
d7	분류하기	classification



- 무작위(randomness)

- 의도를 가지고 선택적으로 하지 않고 무작위로 데이터를 모으는 지혜다. 컴퓨터로 무작위를 어떻게 흉내낼 수 있는지, 컴퓨터가 흉내내는 무작위의 한계는 무엇인지 등을 익힌다.

- 확률(probability)

- 상대적인 빈도로서 확률을 계산하는 지혜다. 확률을 수리적으로 계산하는 방법뿐 아니라 컴퓨터가 시뮬레이션을 통해서 계산하는 방법 등을 익힌다.

- 두루살펴보기(exploration/sketches)

- 모은 데이터의 성질을 개괄적으로 살피는 지혜다. 데이터에 어떤 성질의 데이터가 얼마나 많은지, 데이터가 가지는 성질을 기준으로 데이터를 줄세웠을 때 중간이나 평균을 알아보고, 테이터들이 중간이나 평균에서 얼마나 멀리 흘어져있는지 등을 살피는 지혜다.

- 비교탐색해서 예측하기(prediction by search)

- 현재와 비슷했던 과거의 역사를 참고해서 앞으로의 미래를 예측하는 지혜다.
- 비유: 그가 어떤 책을 구입했다. 같은 책을 구입한 다른 독자들이 읽은 다른 책도 그에게 추천한다. 그도 그 책을 좋아할 것으로 예측하기 때문이다.

- 경향파악해서 예측하기(prediction by modeling)

- 어떤 문제의 예제와 답을 충분히 많이 봐왔다면 새로운 예제가 나오면 얼추 답을 맞출 수 있는 지혜다.
- 비유: “서당개 5만년”이면 똑똑한 일을 하는 서당개가 나올 수 있다. 주입식으로 예제와 답을 많이 익히면 새로운 예제의 정답을 내놓는 것이 가능하다. 영어를 한국어로 번역하는 서당개는 영어문장과 한국어 번역문을 수 없이 보면 어느 정도 가능해진다.

- 원인헤아리기(abduction)

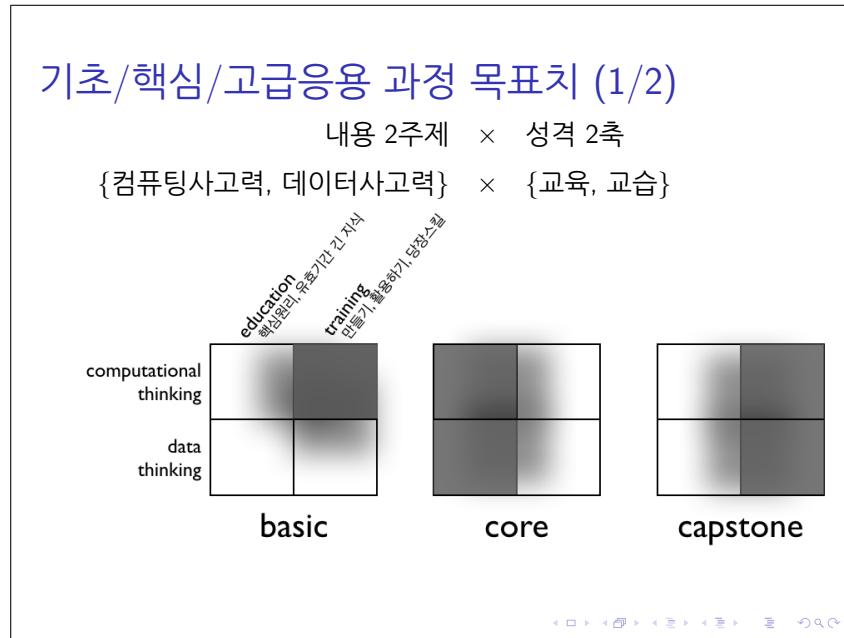
- 데이터를 보고 그런 데이터가 나오게 된 가장 그럴듯한 원인을 헤아리는 지혜다.

- 비유: 반복된 모의고사 시험점수 데이터를 가지고 학생들의 실력 등수를 헤아린다. 반복된 국가대표 축구팀들의 전적을 가지고 축구 국가대표팀의 실력 등수를 헤아린다. 지금까지의 카톡 문자들을 보고 상대방이 나를 좋아하는지를 헤아린다.

- 분류하기(classification)

- 데이터들을 분류할 수 있는 기준을 찾아서 그 기준에 따라 분류하는 지혜다. 새로 출현하는 데이터를 그런 기준에 따라 분류하는 지혜다.
- 비유: 지구상의 꽃들을 겉모양 기준을 세워서 분류한다. 색깔별로, 피는 시기별로, 꽃잎의 갯수별로 등등.

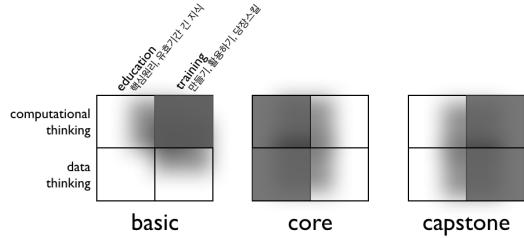
## 5 신규 교과목 목표수준



세 과목의 목표수준은 위의 4분면 도형의 음영으로 표현된다. 4분면은 과목내용 2주제(컴퓨팅 사고력, 데이터 사고력)와 과목성격 2축(교육, 교습)이 만들어 내는 영역이다: 컴퓨팅 사고력 교육, 컴퓨팅 사고력 실습, 데이터 사고력 교육, 데이터 사고력 실습.

검은색 음영부분은 위의 4분면에서 각 단계의 과목이 집중하는 부분을 표현한다. 흐릿한 구름 느낌의 음영은 집중은 아니지만 어느정도 다루는 부분을 표현한다.

## 기초/핵심/고급응용 과정 목표치 (2/2)



**기초:** 두려움극복 흥미유발 별거아니네 오호

**핵심:** 원리파악 흥미유발 알겠다이제 오와

**고급응용**: 어디한번 흥미유발 이런거구나 와아

## 열쇠 = 문제 중심



- 기초 과목

- 컴퓨팅 사고력 실습에 집중한다. 컴퓨팅 사고력 교육도 살짝 커버 한다. 데이터 사고력 실습도 살짝 커버한다.
  - 컴퓨팅 사고력 실습에 우선 집중하면서 학생들이 컴퓨터에 대한 막연한 두려움을 극복할 수 있도록 돋는다. 그래서 컴퓨터를 도구로 이용해서 문제를 해결한다는 것에 대한 자신감을 가지도록 한다. 비유하면, 학생들이 “별거아니네”라는 느낌이 과목을 수강하면 남을 수 있도록 한다.

## • 핵심 과목

- 컴퓨팅 사고력과 데이터 사고력 교육에 집중한다. 컴퓨팅 사고력과 데이터 사고력의 실습도 살짝 커버한다.
  - 컴퓨팅 사고력과 데이터 사고력의 원리를 파악해서 전달하는데 집중한다. 비유하면, 학생들이 과목을 수강한 후, 핵심 원리를 익힌덕

분에 “알겠다이제”라고 느낄 수 있도록 한다. 실습은 원리를 이해시키는 데 필요한 만큼으로 규모를 관리한다.

- 고급 과목

- 컴퓨팅 사고력과 데이터 사고력의 실습에 집중한다. 컴퓨팅 사고력과 데이터 사고력의 교육도 살짝 커버한다.
- 컴퓨팅 사고력과 데이터 사고력의 실습을 규모있게 해 보는데 집중한다. 규모가 크고 실제적인 실습을 해 보면서, 컴퓨팅 사고력과 데이터 사고력의 가능성과 함께 한계에 대해서도 경험할 수 있도록 한다. 비유하면, 학생들이 과목을 수강한 후, 기술의 강점과 약점을 모두 파악해서 “이런거구나”라고 느낄 수 있도록 한다. 컴퓨팅 사고력과 데이터 사고력의 원리는 상기시키는 정도로 한다.

- \* 공통의 방도

- 각 단계마다 위와 같은 목표수준을 달성하는 방법은 각 단계의 과목 공히 문제중심(problem-centered, real-world relevance)으로 운영하는 것이다. 각 목표수준에 적합한 구체적인 문제를 발굴해서 그 문제를 해결하는 과정이 중심이 되도록 기초/핵심/고급 과목을 만든다.
- 6절에 이러한 과목 운영 예시가 있다.

## 5.1 기초과정 목표치 세부

# 기초(basic) 과정 목표치 세부

기초:	두려움극복 원리파악 고급용어: 어디한번	흥미유발 흥미유발 흥미유발	별거아니네 알겠디야제 이런거구나	오후 오와 외야
핵심:				
고급용어:				

- ▶ 문제중심: 후보문제중에 1 ~ 3개 선택, “기초버전” 진행.
- ▶ 교습
  - ▶ 개발경험: 작성한 코드가 문제해결의 일부 부품되도록
  - ▶ 개발규모: ~1KLoC (Python)
  - ▶ 활용경험: 기존 sw를 활용해서 부품을 작성
- ▶ 교육
  - ▶ 컴퓨팅사고력: 3 ~ 6/12개 부품을 코드작성에 적용
  - ▶ 데이터사고력: 0 ~ 2/7개 부품을 코드작성에 적용

- 한 학기동안 후보문제중에서 1~3개를 선택한다.
  - 그 후보문제들의 완전한 버전보다는 기초 버전을 고안해서 사용한다.
  - 학생들은 그렇게 각색한 문제 1~3개를 해결하는 소프트웨어들을 만들게 된다.
  - 조교와 교수는 문제를 해결한 해답 소프트웨어를 미리 만들어 놓는다.
  - 조교와 교수는 해답 소프트웨어에서 학생들이 한 학기 동안 담당할 부품들을 파악한다.
  - 학생들은 그 부품들을 메꾸는 과정이 실습과정이 된다.
    - 실습언어를 Python으로 했을 경우 학생들이 한 학기 동안 약 1000 줄 정도 규모의 코드를 써보도록 한다.
    - 실습중에 기존에 작성되었는 코드(library)를 활용하도록 한다.

- 학생들을 컴퓨팅 사고력 12 개념 중에서 3~6개 정도를 익히게 된다.
- 학생들은 데이터 사고력 7 개념 중에서 0~2개 정도를 익히게 된다.

## 5.2 핵심과정 목표치 세부

**핵심(core) 과정 목표치 세부**

기초:	두려움극복	흥미유발	별거아니네	오후
<b>핵심:</b>	원리파악	흥미유발	알겠다이제	오와
고급능동:	어디한번	흥미유발	이런거구나	와야

- ▶ 문제중심: 후보문제중에 1 ~ 3개 선택, “핵심버전” 진행.
- ▶ 교습
  - ▶ 개발경험: 작성한 코드가 문제해결의 일부 부품되도록
  - ▶ 개발규모: ~3KLoC (Python)
- ▶ 교육
  - ▶ 컴퓨팅사고력: 7 ~ 10/12개 부품을 코드작성에 적용
  - ▶ 데이터사고력: 2 ~ 4/7개 부품을 코드작성에 적용

- 한 학기동안 후보문제중에서 1~3개를 선택한다.
- 그 후보문제들의 완전한 버전보다는 핵심 버전을 고안해서 사용한다.  
핵심 버전은 기초 버전보다 다뤄야 할 세부사항이 확장된 것이다.
- 학생들은 그렇게 각색한 문제 1~3개를 해결하는 소프트웨어들을 만들게 된다.
- 조교와 교수는 문제를 해결한 해답 소프트웨어를 미리 만들어 놓는다.
- 조교와 교수는 해답 소프트웨어에서 학생들이 한 학기 동안 담당할 부품들을 파악한다.
- 학생들은 그 부품들을 메꾸는 과정이 실습과정이 된다.

- 실습언어를 Python으로 했을 경우 학생들이 한 학기 동안 약 3000 줄 정도 규모의 코드를 써보도록 한다.
- 실습중에 기존에 작성되었는 코드(library)를 활용하도록 한다.
- 학생들을 컴퓨팅 사고력 12 개념 중에서 7~10개 정도를 익히게 된다.
- 학생들은 데이터 사고력 7 개념 중에서 2~4개 정도를 익히게 된다.

### 5.3 고급과정 목표치 세부

**고급응용(capstone) 과정 목표치 세부**

기초:	두려움극복	흥미유발	별거아니네	오호
핵심:	원리파악	흥미유발	알겠다이제	오와
고급응용:	어디한번	흥미유발	이런거구나	와아
<b>▶ 문제중심:</b>	후보문제중에 1 ~ 3개 선택, “고급버전” 진행.			
<b>▶ 교습</b>	<ul style="list-style-type: none"> <li>개발경험: 작성한 코드가 문제해결의 일부 부품되도록</li> <li>개발규모: ~5KLoC (Python)</li> </ul>			
<b>▶ 교육</b>	<ul style="list-style-type: none"> <li>컴퓨팅사고력: 12/12개 부품을 코드작성에 적용</li> <li>데이터사고력: 2 ~ 3/7개 부품을 코드작성에 적용</li> </ul>			

- 한 학기동안 후보문제중에서 1~3개를 선택한다.
- 그 후보문제들의 완전한 고급 버전을 고안해서 사용한다. 고급 버전은 핵심 버전보다 다뤄야 할 세부사항이 확장된 것이다.
- 학생들은 그렇게 각색한 문제 1~3개를 해결하는 소프트웨어들을 만들게 된다.

- 조교와 교수는 문제를 해결한 해답 소프트웨어를 미리 만들어 놓는다.
- 조교와 교수는 해답 소프트웨어에서 학생들이 한 학기 동안 담당할 부품들을 파악한다.
- 학생들은 그 부품들을 메꾸는 과정이 실습과정이 된다.
  - 실습언어를 Python으로 했을 경우 학생들이 한 학기 동안 약 5000 줄 정도 규모의 코드를 써보도록 한다.
  - 실습중에 기존에 작성되었는 코드(library)를 활용하도록 한다.
  - 학생들을 컴퓨팅 사고력 12 개념을 모두 동원하게 된다.
  - 학생들은 데이터 사고력 7 개념 중에서 2~3개 정도를 익히게 된다.

## 6 문제중심 교과과정 예시

예를들어, 문제 “기초교육원 교양과목 수강규칙 문답 챗봇을 만들자”를 중심으로 기초/핵심/고급 교과목이 운영되는 예시는 다음과 같다.

위 문제의 기초버전/핵심버전/고급버전이 각각 기초/핵심/고급 교과목의 문제중심 운영에 사용된다.

### 6.1 문제중심 기초버전 예시

#### 문제중심 기초버전 예시

문제: 기초교육원 교양과목 수강규칙 문답 챗봇을 만들자

- ▶ 글쓰기 과목 수강규칙을 내 전공의 경우 파악하기
- ▶ 알아본 수강규칙을 형식을 정해서 표현해보기
  - ▶ 데이터 표현/정리/구현
  - ▶ 데이터베이스 활용
- ▶ 질문 형식을 한정하기 (finite & fixed)
  - ▶ 데이터 표현/구현 (풀다운 메뉴)
  - ▶ 표현력이 상당히 제한된 질문언어 정의하기
- ▶ 질문마다 수강규칙표의 답을 찾는 방법 고안하기
  - ▶ 질문언어 처리기 만들기
  - ▶ 논리 구현 (예/아니오 그래프, 선택결정 그래프)
- ▶ 사고력: (c1, c3, c6, c8, c11, c12) + ()



- 문제의 기초버전:

- 기초교육원 교양과목중에서 글쓰기 과목들로 한정한다.
  - 말로 주고받는 문답 챗봇이 아니라 한정된 메뉴로 질문을 구성하도록 한정한다.
  - 답안은 한정된 형식으로 화면에 프린트되도록 한정한다.
- 글쓰기 수강규칙을 각 전공별로 파악해서 과목 수강 선후 규칙등을 표현해 보도록 한다.

- 각각의 규칙 데이터를 정리하고 표현하고 컴퓨터 데이터로 구현해 본다.
- 이러한 데이터를 데이터베이스 시스템을 활용해서 저장해 본다.
- 규칙들을 하나의 예/아니오 그래프(decision tree)로 표현해 본다.
- 질문을 표현하는 한정적인 방안을 고안한다.
  - 풀다운 메뉴들에서 질문을 선택해서 조합하는 방식을 고안한다.
- 입력된 질문에 대해서 수강규칙을 보고 답하는 과정을 고안한다.
  - 예/아니오 그래프(decision tree)로 표현한 수강규칙을 따라 질문의 답을 추적하는 과정을 고안한다.
- 컴퓨팅 사고력: c1, c3, c6, c8, c11, c12 등을(4.1절) 익히게 된다.
- 데이터 사고력: 없음.

## 6.2 문제중심 핵심버전 예시

### 문제중심 핵심버전 예시

문제: 기초교육원 교양과목 수강규칙 문답 챗봇을 만들자

- ▶ 글쓰기와 수학과목 수강규칙을 전공별로 알아보기
- ▶ 알아본 수강규칙을 형식을 정해서 표현해보기
  - ▶ 데이터 표현/정리/구현
  - ▶ 데이터베이스 시스템 소개/활용
- ▶ 임의의 질문을 표현하는 형식 정하기 (quasi-infinite)
  - ▶ 데이터 표현/반복/재귀
  - ▶ 표현력이 한정된 입력언어 정의하기
- ▶ 질문마다 수강규칙표의 답을 찾는 방법을 고안하기
  - ▶ 질문언어 처리기 만들기
  - ▶ 중규모 논리 구현 (예/아니오 그래프, 선택결정 그래프)
  - ▶ 비교탐색해서 추천하기/원인헤아리기
- ▶ 사고력: (c1, c2, c3, c4, c6, c7, c8, c9, c11, c12) + (d1, d2, d3, d4, d6)



- 문제의 핵심버전:

- 기초교육원 교양과목중에서 글쓰기와 수학과목으로 한정한다.
  - 말로 주고받는 문답 챗봇이 아니라 한정된 문법의 텍스트로 질문을 구성하도록 한정한다.
  - 답안은 한정된 형식으로 화면에 프린트되도록 한정한다.

- 글쓰기와 수학과목의 수강규칙을 각 전공별로 파악해서 과목 수강 선후 규칙등을 표현해 보도록 한다.

- 규칙 데이터를 정리하고 표현하고 컴퓨터 데이터로 구현해 본다.
  - 이러한 데이터를 데이터베이스 시스템을 활용해서 저장해 본다.
  - 규칙들을 하나의 예/아니오 그래프로 표현해 본다(decision tree).

- 질문을 표현하는 한정적인 방안을 고안한다.

- 제한된 문법의 질문언어를 정한다.
  - 정해진 문법으로 표현되는 질문문장을 입력으로 받아 문장의 구조(의미)를 파악하는 처리기를 구성해 본다.

- 의미를 파악한 질문문장에 대해서 수강규칙을 보고 답하는 과정을 고안 한다.

- 예/아니오 그래프로 표현한 수강규칙을 따라 질문의 답을 추적하는 과정을 고안한다.
  - 비교탐색 추천하기를 해 본다: 비슷한 질문을 한 학생이 수강한 다른 과목을 질문한 학생에게 추천해 주는 시스템의 부품을 만들어 본다.
  - 원인해아리기를 해 본다: 비슷한 질문을 한 학생들의 관심분야와 전공등을 파악하고 이를 기반으로 질문한 학생의 관심분야와 전공을 예측하는 시스템의 부품을 만들어 본다.

- 컴퓨팅 사고력: c1, c2, c3, c4, c6, c7, c8, c9, c11, c12 등을(4.1절) 익히게 된다.
- 데이터 사고력: d1, d2, d3, d4, d6 등을(4.2절) 익히게 된다.

### 6.3 문제중심 고급버전 예시

#### 문제중심 고급버전 예시

[문제: 기초교육원 교양과목 수강규칙 문답 챗봇을 만들자]

- ▶ 논리적으로 해결하기
  - ▶ 글쓰기/영어/수학/과학 교양과목 수강규칙을 전공별로 알아보기
  - ▶ 수강규칙을 형식을 정해서 표현해보기 (데이터 표현/정리/구현)
  - ▶ 임의의 질문을 표현하는 형식 정하기 (제거, 표현력 높은 일학연어 정의)
  - ▶ 질문마다 수강규칙표의 답을 찾는 방법을 고안하기
    - ▶ 질문언어 처리기 만들기 (번역/제거)
    - ▶ 중규모 논리 구현 (예/아니오 그래프, 선택결정 그래프)
    - ▶ 문의가 자연어 말이나 글로 입력되게 하기 (뉴럴넷)
- ▶ 딥러닝으로 해결하기
  - ▶ 학생의 특성/경향들을 입력하면 1학년 수강표를 제안해주는 AI만들기
  - ▶ 축적한 1학년 수강 데이터를 이용하기 (뉴럴넷)
- ▶ 사고력: (c1~c12) + (d5, d7)



- 문제의 고급버전:

- 기초교육원 교양과목 모두(글쓰기/영어/수학/과학 과목)를 대상으로 한다.
- 자연어 글이나 말로 문답하는 챗봇을 만들도록 한다.
- 모든 교양과목의 수강규칙을 모든 전공별로 파악해서 과목 수강 선후 규칙등을 표현해 보도록 한다.
  - 규칙 데이터를 정리하고 표현하고 컴퓨터 데이터로 구현해 본다.
  - 이러한 데이터를 데이터베이스 시스템을 활용해서 저장해 본다.

- 모든 규칙을 예/아니오 그래프로 표현해 본다(decision tree).
- 질문은 자연어 문장이나 소리로 입력할 수 있도록 한다.
  - 질문 문장의 표현을 특정 문법으로 제한하지 않고, 한국어 문장을 쓰거나 말로 입력하도록 한다.
- 입력된 질문에 대해서 수강규칙을 보고 답하는 과정을 고안한다.
  - 예/아니오 그래프로 표현한 수강규칙을 따라 질문의 답을 추적하는 과정을 고안한다.
  - 비교탐색 추천하기를 해 본다: 비슷한 질문을 한 학생이 다른 과목은 무엇을 들었는지 추천해 주는 시스템의 부품을 만들어 본다.
  - 원인해아리기를 해 본다: 비슷한 질문을 하는 학생들의 관심분야와 전공등에 대해서 파악하고 이를 기반으로 질문자의 관심분야와 전공을 예측하는 시스템의 부품을 만들어 본다.
- 기계학습으로 학생들의 전공과 특성을 입력하면 교양과정 1학년 수강표를 제안해 주는 인공지능 시스템을 만들어 본다.
  - 기초교육원에서 축적한 1학년 수강 데이터를 기계학습 데이터로 활용한다.
- 컴퓨팅 사고력: c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12 등을(4.1절) 익히게 된다.
- 데이터 사고력: d5, d7 등을(4.2절) 익히게 된다.

## 7 교수/조교/강의자원 수급

## 7.1 수업진행방식

## 교수/조교/강의자원 수급 (1/2)

- ▶ 교수: 강의(60%) + 녹화활용(40%, flipped learning),  
문제 시나리오 작성, 부품문제 구성
    - ▶ 전공교수가 첫 3 ~ 5년은 담당: 강의자료 숙성후  
강의교수에 전수
  - ▶ 조교/학부생 튜터: 실습 현장지도
  - ▶ 실습장비: 노트북, 칠판, 실습 인프라sw,  
`gym.openai.com` 활용
  - ▶ 자료보급: 책출판/YouTube (서울대 리더십)

#### • 교수 강의

- 현장강의와 녹화된 동영상을 6:4 비율로 배합한다:  
현장강의와 동영상강의를 활용한 거꾸로학습(flipped learning)을 약 6:4 비율로 진행한다. 100% 거꾸로학습 방식으로 진행하면 학생들에게 부담이 크고 교육효과도 적을 수 있다.
  - 학생은 교수를 강의현장에서 만나야한다:  
강의현장에서 즉흥적으로 주고받는 학생과의 문답과 그때그때 다른 강의현장에서 직접 학생들과 만나는 교육효과를 무시할 수 없다. 서울대 입학생들에게는 이런 현장감있는 강의를 통해서 교육효과를 최대화하게 된다.
  - 문제정의 및 실습부품 파악:

학기 시작전에 교수는 문제중심 방식으로 진행할 문제들을 고안하고, 조교와 함께 그 문제를 해결한 소프웨어를 완성하며, 그 소프트웨어의 부품중에서 학생들에게 부과할 실습 파트를 정한다.

- 첫 3 ~ 5년은 전공교수가 강의:

전공교수가 3 ~ 5년 강의를 통해서 강의자료가 정비된 후에 강의 전담 교수에게 전수한다.

- 조교/학부생 튜터

- 조교와 함께 학부생 튜터를 모집해서 실습현장에서 학생들을 돋도록 한다.

학부생 튜터는 이전 수강생 중에서 우수했던 학생을 찾아 스카웃/모집하도록 한다.

- 조교는 학기전에 담당교수와 문제중심 방식으로 진행할 문제들의 해답을 모두 완성한다.

- 실습장비

- 학생들 개개인의 노트북으로 코딩하도록 한다.

- 코딩 실습 환경은 학생들 노트북의 운영체제 종류(MS Windows, Mac iOS, Linux 등)에 모두 적용할 수 있는 것으로 한다. 웹브라우저를 활용하는 경우에도 모든 웹브라우저(Chrome, Internet Explorer, Firefox, Opera 등)에서 사용할 수 있는 것을 선택한다.

- gym.openai.com 등 이미 잘 준비된 인공지능 프로그래밍 실습 환경을 활용한다.

- 강의 자료 보급

- 수년간의 강의를 통해서 축적하고 정비한 강의자료는 장벽없이 교내외 누구에게나 무상으로 보급할 수 있도록 한다.

- 강의내용을 책(종이와 디지털)과 YouTube 동영상으로 제작한다.

## 교수/조교/강의자원 수급 (2/2)

#### ▶ 학점체계

- ▶ 기초 2:2:0, 핵심 3:3:0, 고급응용 3:3:0
  - ▶ 강의시간에 강의 및 실습 병행

#### ▶ 성적부과

- ▶ S/U 기본, 학생이 A~F 선택
  - ▶ 또는, 전공별로 분반 성적부과



- 학점체계는 기초 2:2:0 (학점: 주 강의시간: 주 실습시간), 핵심 3:3:0, 고급 3:3:0 으로 한다.
    - 실습은 강의 시간에 병행한다.
  - 성적은 S/U를 기본으로 한다.
    - 원하는 학생에게는 A~F 방식으로 성적을 준다.
    - 학생들이 컴퓨터 프로그래밍 경험이 다르면 불공정한 과정이 될 수 있다. 전공별로 분반해서 성적을 부과하는 것도 고려한다.
    - 경험있는 학생이 쉽게 강의를 수강하고 쉽게 성적을 받으려는 경향이 있다면, 그러한 학생들의 경향을 역이용해서 강의에 긍정적으로 활용하는 방안이 가능하다 (7.2.2절 참고).

## 7.2 실습 인프라 구성

### 7.2.1 실습 코드 자동 채점기

#### 실습 인프라 구성 (1/2)

- ▶ 웹 기반, OS/웹브라우저 중립
- ▶ 교내 개발 후 오픈소스화: 국내외 보급
  - ▶ 상용: Codenvy, SourceLair, Koding, Codeanywhere
  - ▶ 오픈소스: Jupyter, Eclipse, Codebox, edu.goorm.io
- ▶ 서울대가 소유해야: 학생 코드 데이터
  - ▶ 코딩 교습법 연구용: 학습과정 데이터 확보
  - ▶ 코드 채점 AI용: 훈련 데이터 확보  
(shallow → deep automation)
- ▶ 방향
  - ▶ 코드 채점: 100% 자동화 (shallow automation) 시스템
  - ▶ 코드 수정돕기: 40% 상부상조 (peer-sourcing) 플랫폼  
(+ 60% 조교/튜터 도움)



- 실습 코드가 옳은지 틀린지를 테스트를 통해 판단하는 몇 개의 입력에 대한 실행결과를 보고 판단하는 시스템은 100% 자동화가 가능하다.
- 서울대가 외주를 통해 개발하도록 한다.
- 유의 사항:
  - 이 정도 자동화에 머물러서는 곤란(shallow automation):  
이런 방식의 자동 채점기는 당장 구현가능한 단순한 자동화인데,  
이 정도의 인프라에 머물러서는 강의효과가 정체될 수 있다.
  - 조교나 튜터 수준의 자동화를 계획해야(deep automation):  
틀린 코드의 경우 어디가 틀렸고, 어떻게 고치면 될지를 학생들  
경우마다 개별적으로 자세히 힌트를 주는 자동 시스템이 필요하다.

학생 개인별 도움을 줄 수 있는 조교와 튜터의 인원이 제한적이기 때문에 더욱 그렇다.

- 이 정도로 지능화된 시스템은 아직 완전 자동화가 불가능하지만, 많은 코드가 축적되면 인공지능기술(기계학습 등)과 소프트웨어기술(프로그램 분석 기술 등)의 조합으로 곧 가능해 질 것이다.
  - 이러한 미래를 선도하려면, 서울대가 학생들이 제출하는 모든 실습 코드를 축적하고 소유하고 있어야 한다.
  - 그런 실습코드는 효과적인 코딩 교육법 연구뿐 아니라, 코드채점과 수정힌트를 자동으로 해주는 인공지능 시스템 개발에 꼭 필요한 자원이 되기 때문이다.
- 단순 자동 채점기를 보완할 인프라로, 수강하는 학생들이 모두 참여하는 “코드 수정돕기 상부상조 플랫폼”이(7.2.2절 참고) 가능하다.
    - 이 플랫폼을 활용해서 조교와 튜터의 부하를 줄이고, 학생들끼리 서로 돋는 환경을 자극할 수 있다.
    - 학생들 사이의 실력/경험 차이를 순기능으로 활용할 수 있는 플랫폼이기도 하다.

### 7.2.2 코드 수정돕기 상부상조 플랫폼

## 실습 인프라 구성 (2/2)

코드 수정돕기 상부상조 (peer-sourcing) 플랫폼 ([crowdgrader.org](http://crowdgrader.org))

- ▶ 학생이 서로서로
  - ▶ 리뷰 하고 + 한 리뷰 평가받고
  - ▶ 리뷰 받고 + 받은 리뷰 평가하고
- ▶ 동기부여 방안: 리뷰 질↑
  - ▶ 받은 리뷰를 평가: 기술면 + 감성면
  - ▶ 리뷰 평가점수 ~ 성적/학기말표창/튜터후보확보
- ▶ 기대 효과
  - ▶ 동료에게 배우기 + 상부상조 정신 진작
  - ▶ 수강생 차이 활용 + 대형강의 조교부족 해소

- 학생들끼리 서로 돋는 플랫폼:

이 플랫폼에서는 조교나 튜터의 역할을 학생들 서로가 하게 된다. 학생들끼리 자신들이 제출한 실습코드의 문제점을 서로 지적해주고 상부상조할 수 있도록(peer-sourcing) 돋는다. 이렇게 대형강의의 조교부족 문제를 해소할 수 있다.

- 그러한 플랫폼으로 crowdgrader.org를 활용할 수 있다.
  - 서울대가 같은 기능의 플랫폼을 외주를 통해 개발할 수도 있다.
  - 상부상조 플랫폼 기능
    - 무작위 리뷰 배정: 학생들은 무작위로 다른 학생들의 코드를 배정 받아 보게되고, 틀린점이나 개선점 혹은 응원메세지 등을 리뷰 코멘트로 보낸다. 누가 누구의 코드를 리뷰했는지는 밝혀지지 않는다 (double-blind).

- 보낸 리뷰를 평가받는다: 학생들은 자신들이 보낸 리뷰를 리뷰받은 학생들로부터 평가받는다.
- 받은 리뷰를 평가한다: 학생들은 자신들의 코드에 대해서 받은 리뷰가 긍정적인 도움이 되었는지를 평가한다.
- 성적에 반영: 리뷰의 질과 빈도를 성적에 반영해서 열심히 리뷰하도록 유도한다.

- 상부상조 플랫폼의 부수효과

- 동료에게 배우기: 학생들은 상대적으로 잘하는 학우의 좋은 답을 보고 배우는 효과가 매우 크다.
- 상부상조 문화 장려: 잘하는 학생은 다른 학우들의 코드를 보고 도와주며 보람을 느낀다. 학생들 사이에 상부상조의 문화가 만들어 진다.
- 수강생사이의 편차 활용: 잘하는 학생이 다른 학우들을 돋는 것이 시스템적으로 장려되기 때문에, 쉽게 학점따는데 수강생 편차가 일방적으로 이용만 되지는 않는다.
- 튜터 발굴 용이: 활발히 좋은 코멘트를 많이 한 학생을 다음 학기 학부생 튜터로 활용할 수 있다.

### 7.2.3 데이터 사고력 실습 환경

- 인공지능 실습 – 기계학습(machine learning)과 딥러닝(deep learning)을 이용한 비교탐색해서 예측하기, 경향파악해서 예측하기, 분류하기, 원인 헤아리기 프로그래밍(probabilistic programming) 등은 그 실행에 일반 노트북으로는 감당하기 힘든 많은 데이터와 계산이 필요하다.
- 따라서 컴퓨팅 용량이 크고 특별하게 갖춰진 기계들이 데이터 사고력 실습에 필요하다. 이런 실습에 필요한 고가의 서버들을 갖추고 공동으로 활용하는 환경이 필요하다.

### 7.3 교과목 이름

#### 과목 이름

엄숙 vs 경쾌

고상 vs 담백

훈련 vs 놀이

- ▶ The Beauty and Joy of Computing

- ▶ 컴퓨팅의 아름다움과 즐거움
- ▶ 처음 만나는 컴퓨팅
- ▶ 컴퓨팅의 즐거움, 즐거운 컴퓨팅, 컴퓨팅 즐기기
- ▶ 달달한 컴퓨팅

- ▶ Thinking in Computation & Data

- ▶ 컴퓨팅 역량키우기: 기초/핵심/고급
- ▶ 컴퓨팅 훈련과 이해: 기초/핵심/고급
- ▶ 컴퓨터로 생각하기: 기초/핵심/고급
- ▶ 컴퓨팅의 도전/이해/응용



- 모두에게 편하게 다가가는 과목명: 과목명은 과정의 의도에 맞추어 전 공불문 남녀 학생 모두에게 편하게 다가갈 수 있도록 해야 한다. 특히 여학생들에게 컴퓨터기술에 대한 흥미가 의도치 않게 사그라들지 않도록 과목명에서부터 세심한 접근이 필요하다.
- 이를 위해서, 건조하고 엄숙하고 기계적인 과목명에서 벗어나서 담백 경쾌하고 위압적이지 않은 이름을 과목명으로 사용하는 방안이 있다.
- 위 슬라이드에서 “The Beauty and Joy of Computing”(UC Berkeley에서의 유사 과목 이름) 꼭지 아래의 이름들이 그러한 예로 찾은 것이다.

## 8 부록: 현황 분석

### 8.1 현재 컴퓨터관련 교양교과목 및 이수규정

현황 분석					
대 영 역 영 역	소영역명	교과목명	개설대학	학점구조 (학점-이론-실습)	비고
학문의 기초	컴퓨터와 정보활용	컴퓨터의 개념 및 실습	인문대학, 경영대학, 공과대학, 사범대학	3-2-2	
		공과대학 의 기본 기초	2-2-0		
		컴퓨터의 기초 과 학제간 융합 학습	3-2-2	2016학년도 1학기 신설	
		과학기술을 위한 컴퓨터 활용 기법	3-2-2	2017학년도	2017학년도 2학기 신설
		컴퓨터의 사고와 실습	3-2-2	2학기 신설	2018학년도 1학기 신설
	학문의 세계	인공지능 입문	인공지능 부	3-2-2	2018학년도 1학기 신설
		정보사회와 자연대학 학	수리과학부 컴퓨터 과학 부	3-3-0	
		에는 세계 인 터넷 보안 과 학제 교류	3-3-0		
		프리아 비데이터 의 이 해	컴퓨터 부	3-3-0	2019학년도 2학기 신설
		산업공학 과	산업공학 과	3-3-0	2019학년도 2학기 신설
선택교 양	선택교 양	정의와 유 형	공과대학 의 기초	3-3-0	2019학년도 1학기 신설
		정의와 유 형	선택교 양		
		정의와 유 형	선택교 양		
		정의와 유 형	선택교 양		

● 컴퓨터 관련 개설 교양교과목: 2019년 2학기 기준으로 학문의 기초, 컴퓨터와 정보 활용, 영역에 5개의 교과목, 학문의 세계, 자연과 기술, 영역에 3개의 교과목, 선택교양 창의와 융합 영역에 1개의 교과목이 개설되어 있다.

대 영 역 영 역	소영역명	교과목명	영역	학점구조 (학점-이론-실습)	비고
공과 대 학	공과대학 각 학과	컴퓨터의 개념 및 실습	학문의 기초	3-2-2	
		컴퓨터의 기초 과학적 사고와 실습	컴퓨터와 정보활용	2-2-0	
	제료 공학 부	컴퓨터의 기초 과학적 사고와 실습	2017학년도 2학기 신설	3-2-2	2017학년도 2학기 신설
		인공지능 입문	2018학년도 1학기 신설	3-2-2	2018학년도 1학기 신설
		컴 퓨 티 과 학부	컴 퓨 티 과 학 이 여는 세계 인 터넷 보안 과 학 교류	3-3-0	2019학년도 2학기 신설
	산업 공학 과	컴 퓨 티 과 학부	학문의 세계 인 터넷 보안 과 학 교류	3-3-0	2019학년도 2학기 신설
		프리아 비데이터 의 이 해	프리아 비데이터 의 이 해	3-3-0	2019학년도 2학기 신설
		선택교 양	선택교 양	3-3-0	2019학년도 1학기 신설
	인문 대 학	컴 퓨 티 의 개념 및 실습	선택교 양		
		경영 대 학	컴 퓨 티 의 개념 및 실습		
선택교 양	선택교 양	경영학과 학 부	경영학과 학 부		
		수학과 학 부	수학과 학 부		

● 컴퓨터와 정보활용 이수규정: 컴퓨터와 정보활용 영역의 교과목을 필수로 지정한 대학은 경영대학, 공과대학, 사범대학(수학교육과, 선택필수)이다.

● 시고와 표현, 외국어는 전대학, 수령적 분석과 주문, 과학적 사고와 실천은 이공계 학생 전부가 필수로 수강하게 되어 있는 것을 감안하면 다른 학문의 기초 영역에 비하여 절대적으로 수강대역이 적다.

## 8.2 현재 컴퓨터관련 교양교과목 기초 통계

대학	학과	학점수	전장학기	이수교과목
경영대학	경영학과	3	1학년	컴퓨터의 개념 및 실습 (2018학년도 신입생부트)
공과대학	재료공학부	2	1-2	컴퓨터의 기초
	전기·정보공학부, 컴퓨터공학부, 건축학과 건축공학전공, 산업공학과, 조선해양공학전과, 에너지자원공학과, 원자력공학과	3	1-1	컴퓨터의 개념 및 실습
	화학생물공학부		1-2	
	건설환경공학부, 기계항공공학부		2-1	
사범대학	수학교육과	2-3		컴퓨터의 기초, 컴퓨터의 개념 및 실습 중 약 1

● 학문의 기초에는 7개의 소영역이 있으며 2018학년도 기준					
소영역	교과목 수	강좌 수	수강인원	평균성적	강의평가
사고와 표현	20	243	5092	3.72	4.41
외국어	191	676	10849	3.45	4.49
외국어 I (영어)	33	297	4577	3.36	4.48
외국어 II (제2 외국어)	158	379	6272	3.50	4.50
수량적 추론	41	230	9076	3.19	4.29
과학적 사고와 실험	69	419	10665	3.14	4.04
컴퓨터와 정보 활용	19	27	1300	3.20	4.02
전체합계	340	1595	36982	3.34	4.28

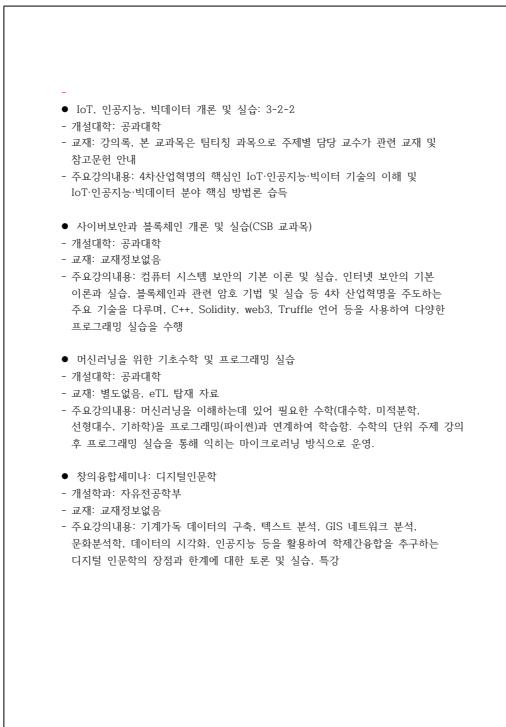
  

● 컴퓨터 관련 교양교과목 요약(2018학년도)					
소영역	교과목 수	강좌 수	수강인원	평균성적	강의평가
학문의 기초 컴퓨터와 정보활용	컴퓨터의 개념 및 실습	23	1100	3.17	4.09
	컴퓨터의 기초	1	102	3.08	3.91
	과학 계산을 위한 컴퓨터 활용	2	64	3.64	2.98
	인공지능 일본	1	34	3.56	3.79
	컴퓨터과학적 사고와 실습	미개설			
학문의 세계 자연과 기술	정보사회의 수학	미개설			
	컴퓨터과학이 '여는 세계'	1	65	3.55	3.81
	인터넷보안과 프로그래밍	1	63	3.13	4.00
선택교양 창의적 융합 전체합계	아버지				
	빅데이터의 이해	1	29	3.41	4.32
		30	1457	3.22	4.02

### 8.3 현재 컴퓨터관련 교양교과목 강의계획서 검토

<p>[학문의 세계: 자연과 기술]</p> <ul style="list-style-type: none"> <li>- 정교사회와 수학: 3-3-0</li> <li>- 개설학과: 자연대학대학 수리과학부</li> <li>- 교재: 별도의 교재 없음</li> <li>- 주요강의내용: 정보사회의 단면을 이루는 여러 가지 기술들을 원대 수학의 관점에서 체험하도록 함. 양자정보와 수학, 수치분석과 딥러닝의 수학, 현대암호학과 개인정보보호의 수학, 위상수학과 데이터 분석 4개의 주제를 다룬.</li> </ul> <ul style="list-style-type: none"> <li>● 컴퓨터과학이 어는 세계: 3-3-0             <ul style="list-style-type: none"> <li>- 개설학과: 공과대학 컴퓨터공학부</li> <li>- 교재: 컴퓨터과학이 어는 세계(이정근, 도서출판 인사이트, 2015)</li> <li>- 주요강의내용: 학생들이 각 분야에서 컴퓨터가 어는 미래를 독창적으로 연출할 수 있도록 컴퓨터 과학기술의 핵심 지식을 습득함. 소프트웨어 기술, 컴퓨터과학을 이용한 전공분야의 시야 확장, 컴퓨터 과학기술의 미래 등 컴퓨터 과학기술의 핵심 지식을 다룬.</li> </ul> </li> <li>● 인터넷 보안과 프라이버시: 3-3-0             <ul style="list-style-type: none"> <li>- 개설학과: 공과대학 컴퓨터공학부</li> <li>- 교재: 컴퓨터 보안과 암호(William Stallings, 그린, 2011), 정전 정보보호개론(이준택 외, 법무사, 2016), 알기쉬운 정보보호개론(히로시 유기, 인피니티 북스, 2017)</li> <li>- 주요강의내용: 경의 내용은 인터넷 구조와 동작, 인터넷 보안, 그리고 인터넷</li> </ul> </li> </ul>	<p>프라이버시 세 영역으로 구성됨. 인터넷 구조는 프로토콜 정의, 인터넷 동작 원리, TCP, IP, DNS, 인터넷 유통 프로토콜들을 다룬며, 인터넷 보안은 보안의 기본 성질 및 개념, 암호화/복호화의 기초, 인증, 보안 장비 등을 다루고, 프라이버시는 인터넷상의 개인 민감정보 유출, 법적 이슈 등을 다룬. 또한 Tor 등 익명화 기법을 포함한 프라이버시 강화 기법, 암호화폐, 블록체인과 디지털 포렌식에 대한 지식을 습득함.</p> <p>[선택교양: 정의와 융합]</p> <ul style="list-style-type: none"> <li>● 빅데이터의 이해: 3-3-0</li> <li>- 개설학과: 공과대학 산업공학부</li> <li>- 실습도구: Matlab</li> <li>- 교재: '빅데이터' 기초: 개념, 동인, 기법", Earl 등, 시그마프레스 2017</li> <li>- 주요강의내용: 빅데이터란 무엇이고, 어떻게 분석하며, 어떻게 활용하는지 공부하고, 실제 데이터를 가지고 실습을 통해 체험한다. 확보 가능한 데이터를 찾아 시작화 및 분석을 통해 인사이드를 도출</li> </ul>
---	---

## 8.4 현재 컴퓨터관련 기타 교과목 및 프로그램



- IoT, 인공지능, 빅데이터 개론 및 실습: 3-2-2  
- 개설대학: 공과대학  
- 교재: 강의록, 본 교과목은 팀티칭 과목으로 주제별 담당 교수가 관련 교재 및 참고문헌 안내  
- 주요강의내용: 4차산업혁명의 핵심인 IoT·인공지능·빅데이터 기술의 이해 및 IoT·인공지능·빅데이터 분야 핵심 방법론 습득
- 사이버보안과 블록체인 개론 및 실습(CSB 교과목)  
- 개설대학: 공과대학  
- 교재: 교재정보없음  
- 주요강의내용: 컴퓨터 시스템 보안의 기본 이론 및 실습, 인터넷 보안의 기본 이론과 실습, 블록체인과 관련 암호 기법 및 실습 등 4차 산업혁명을 주도하는 주요 기술을 다루며, C++, Solidity, web3, Truffle 언어 등을 사용하여 다양한 프로그래밍 실습을 수행
- 머신러닝을 위한 기초수학 및 프로그래밍 실습  
- 개설대학: 공과대학  
- 교재: 별도없음, oTL 탐색 자료  
- 주요강의내용: 머신러닝을 이해하는데 있어 필요한 수학(대수학, 미적분학, 선형대수, 기하학)을 프로그래밍(파이썬)과 연계하여 학습함. 수학의 단위 주제 강의 후 프로그래밍 실습을 통해 익히는 마이크로러닝 방식으로 운영.
- 철의융합세미나: 디지털인문학  
- 개설학과: 자유전공학부  
- 교재: 교재정보없음  
- 주요강의내용: 기계가독 데이터의 구축, 텍스트 분석, GIS 네트워크 분석, 문화분석학, 데이터의 시각화, 인공지능 등을 활용하여 학제간융합을 추구하는 디지털 인문학의 창정과 한계에 대한 토론 및 실습, 특강

## 8.5 컴퓨터의 개념 및 실습 강의평가 주관식 답변

<p>[인문대: 사례 수 (약 20 개)]</p> <p>좋았던 점</p> <ul style="list-style-type: none"> <li>● 교과목</li> <li>- 프로그래밍의 개념을 배우기에 적합했다.</li> <li>- 개인적으로 마지막 객체와 클래스를 다루는 부분에서는 이해도가 많이 떨어졌지만 파이썬 입문자에게 적절한 수업이었다고 생각한다. 파이썬이라는 품은 도구적으로 가르치는 수업이 아닌 일반적인 프로그램에 대한 기초 개념을 쌓을 수 있는 수업이다.</li> <li>- 컴퓨터 프로그래밍에 관심이 많아 해당 분야를 다루는 수업을 찾아다녔으나 전부 공대 전공 수업이거나, 전공분야가 아니면 수강할 수 없도록 되어있었다. 그런데도 중 인문계열 학생을 위한 수업을 찾게 되었다. 한 학기동안 전체적으로 만족 스러운 수업이었다. 한 학기 동안 수업을 듣으면서 파이썬 언어를 이정도까지 이해하고 사용할 수 있다는 것이 놀라웠다.</li> <li>- 단시간에 많은 내용을 배울 수 있어서 좋았다.</li> <li>- 파이썬을 이용해서 다양한 문제를 해결하는 것이 재밌었다.</li> <li>- 재미있었다. 실용적인 기법들을 배울 수 있었다.</li> <li>- 프로그래밍에 대해 확실히 알 수 있는 수업이었다. 파이썬 역량을 확실히 키워주었다.</li> <li>- 프로그래밍, 코딩 관련 진로를 고민하는 사람들에게 맛보기로 좋은 전공택시용 강의다. 짧은 시간 내에 파이썬 수박 걸맞기하는 느낌인데 수박의 모든 표면을 할은 듯 해 좋았다.</li> <li>- 코딩에 대해 깊이 있게 알 수 있었다.</li> </ul> <p>● 교수법</p> <ul style="list-style-type: none"> <li>- 처음 프로그래밍을 접하는 학생들에게 거부감없이 잘 따라가도록 지도해주셨다.</li> <li>- 차근차근 가르쳐주셔서 좋았다.</li> <li>- 듣으면 들을수록 흥미가 생기는 수업이었다. 예전도 많이 해주셨다.</li> <li>- 교수님이 열정이 넘치시고 강의력이 좋았다. etd 질문응답 개시판이 활발하게 운영되며 학생에 큰 도움이 되었다. 수업에서 많이 열어간다.</li> <li>- 수업 내용이 체계적이고, 실제로 사용할 수 있는 실용적인 데이터 처리 방법 등에 대해서 더듬었다.</li> <li>- 많이 배워서 능력이 향상되었다. eTL 질문개시판을 매우 잘 활용하였다.</li> <li>- 파이썬을 처음 배울때도 불구하고 이해할 수 있게, 상세히 잘 설명해주셨다.</li> <li>- 과제의 종류가 프로그래밍을 익히는 데 효과적인 것들이었다고 생각한다.</li> </ul>	<p>개선이 필요한 점</p> <ul style="list-style-type: none"> <li>● 교과목</li> <li>- 처음 프로그래밍을 접하는 이들이 따라갈 수 없는 내용인 것 같다.</li> <li>- 실습에서 중간중간에 갑자기 어려운 내용이 등장해서 따라가기가 다소 힘들기도 했습니다.</li> <li>- 학생의 이해도를 점검하는 팀자가 조금 더 많아졌으면 좋겠다.</li> <li>- 인문계 강의의지만 문이과 구분이 안 되어있어 학생들간에 기본지식 차이가 많이 난다.</li> </ul> <p>● 교수법</p> <ul style="list-style-type: none"> <li>- 실제 코딩 과정을 조금 더 자세히 설명해 주시면 좋겠다.</li> <li>- 나이도에 비해 교수님의 설명이 너무 빠르거나 부족해서 초보자에게는 따라가기 힘든 수업이었다.</li> <li>- 조금 줄여라.</li> <li>- 개체지향 프로그래밍 할때 실습이랑 이론이랑 섞어서 하면 딜 풀릴 것 같다.</li> <li>- 좀 더 차근차근 알려주었으면 좋겠고. 다양한 걸 살펴해봤으면 좋겠다.</li> </ul> <p>● 과제 및 시험</p> <ul style="list-style-type: none"> <li>- 처음 프로그래밍을 하는 입장에서 후반부 강의와 과제들은 조금 어려운 측면이 있다. 중간고사와 있어서 다른 과목 학습에 살짝 부담이 된다. 과제 내용은 좋지만 가이드라인이나 헌트 등을 더 많이 제공해주시면 더 수월하게 할 수 있을 것 같다.</li> <li>- 로드가 매우 많은 편이다. 매주 과제가 있어 부담이 될 수 있다.</li> <li>- 과제 나이도가 너무 높았다.</li> <li>- 과제가 많다.</li> </ul> <p>● 기타</p> <ul style="list-style-type: none"> <li>- 질문개시판에 vote이거나 고민트 기능도 있으면 좋겠다. 질문개시판 첫글자 쓰기전에 험연기 변환이 불편하다.</li> </ul> <p>8.5.2 경영대: 사례 수 (약 70개)</p> <p>좋았던 점</p> <ul style="list-style-type: none"> <li>● 교과목</li> <li>- 실용적인 프로그래밍 수업을 들을 수 있어서 좋았다.(13)</li> <li>- 프로그래밍, 코딩에 대해 배우는 것이 흥미로웠다.(10)</li> </ul>
<p>● 교수법</p> <ul style="list-style-type: none"> <li>- 교수님이 수업을 질적으로 해주셔서 좋았다.(4)</li> <li>- 설명을 꼼꼼히 해주셨다.(10)</li> <li>- 실습을 하는 것이 좋았다.(7)</li> <li>- 교수님이 친절하다.(9)</li> <li>- 책, ppt 자료를 많이 준비해 주셔서 좋았다.(2)</li> </ul> <p>● 과제 및 시험</p> <ul style="list-style-type: none"> <li>- 과제를 통해 실력을 향상하고 성장을 할 수 있어 좋았다.</li> </ul> <p>개선이 필요한 점</p> <ul style="list-style-type: none"> <li>● 교수법 및 교과목</li> <li>- 설명이 불친절했다. 뺐었다.(14)</li> <li>- 프로그래밍에 대해 잘 모르는 학생들은 따라가기 힘들었다.(11)</li> <li>- 주입시간이 4시간 통강으로 너무 길었다.(7)</li> <li>- 실습이 더 많았으면 좋겠다.</li> <li>- 수업 자료가 잘 안보이는 경우가 있었다.</li> <li>- 지루했다.</li> <li>- 충당 평수로 들어야 한다는 점이 컴퓨터에 흥미가 없는 나에게는 별로였다.</li> </ul> <p>● 과제</p> <ul style="list-style-type: none"> <li>- 과제 나이도가 높았다.(3)</li> <li>- 과제가 많다.(5)</li> </ul> <p>● 시험 및 퀴즈</p> <ul style="list-style-type: none"> <li>- 퀴즈 및 시험문제가 어렵다.(3)</li> <li>- 퀴즈가 너무 많았다.(2)</li> <li>- 기말고사때 지엽적인 지식에 관한 문제들이 다수 있었다.(2)</li> <li>- 퀴즈가 어렵다.</li> </ul> <p>● 시험 때 코딩을 좋아해 하는 방식은 비합리적이다.</p> <p>[사범대: 사례 수 (약 30개)]</p> <p>좋았던 점</p> <ul style="list-style-type: none"> <li>● 교과목</li> <li>- 컴퓨터에 관해 많은 지식을 얻었고, 워드, 피피티 등 실용적인 것을 배웠다.(10)</li> </ul> <p>● 교수법</p> <ul style="list-style-type: none"> <li>- 교수님이 복습을 많이 해주시고 설명을 잘해주시다.(6)</li> <li>- 교수님이 자세한 설명, 친절하시다.(10)</li> </ul>	<p>개선이 필요한 점</p> <ul style="list-style-type: none"> <li>● 교수법 및 교과목</li> <li>- 설명이 부족했다.(6)</li> <li>- 프로그래밍 언어에 조금 더 초점을 맞추었으면 좋겠다.(2)</li> <li>- 실질적으로 필요한 내용인지 모르겠다.(2)</li> <li>- 수업 시간이 너무 이른시간이었다.</li> <li>- 수업이 너무 좋리다.</li> </ul> <p>● 실습</p> <ul style="list-style-type: none"> <li>- 실습에서 워드/엑셀/파워포인트 알고 C언어나 파이썬 같은 거 배웠으면 좋겠다.(3)</li> <li>- 실습 시간에 수업을 제대로 듣지 않고 미리 과제를 하는 등 수업 참여도가 떨어지는 학생들이 대부분이다.</li> <li>- 실습 시간에 수행자료의 설명은 부족한데 조교님이 각자 봄주시다 보니 2명뿐이라 미리는 경우가 찾았고, 베껴거리시는 경우가 다소 많았다.</li> </ul> <p>● 시험 및 과제</p> <ul style="list-style-type: none"> <li>- 시험문제가 너무 암기위주다.(2)</li> <li>- 중간고사와 기말고사 시험범위의 편차가 너무 크다. 중간고사 범위를 늘리고 기말고사 범위를 줄이면 좋겠다.(2)</li> <li>- 중간고사 때 문제 형식이 너무 쓰는 게 많고 컴퓨터와 상관없어 보이는 파스칼의 원리, 파스칼의 삼각형 등이 시험에 나와서 당황했다.</li> </ul>