

# 컴퓨터공학과 소개

김지웅

인하대학교 컴퓨터공학과

[jieungkim@inha.ac.kr](mailto:jieungkim@inha.ac.kr)

# 연사 소개

**이름** Jieung Kim (김지응)

**소속** 조교수, 인하대학교 컴퓨터공학과 (2022.09~)

**이력** 성균관대학교 컴퓨터공학과 학사 (2009)

카이스트 전산학과 석사 (2011)

예일대학교 전산학과 박사 (2019)

예일대학교 박사후연구원 (2019.06 ~ 2020.04)

구글 리서치 (서울), Personal AI 팀 (Cerebra), 리서치 엔지니어 – privacy & security  
(2020.05 ~ 2022.02)

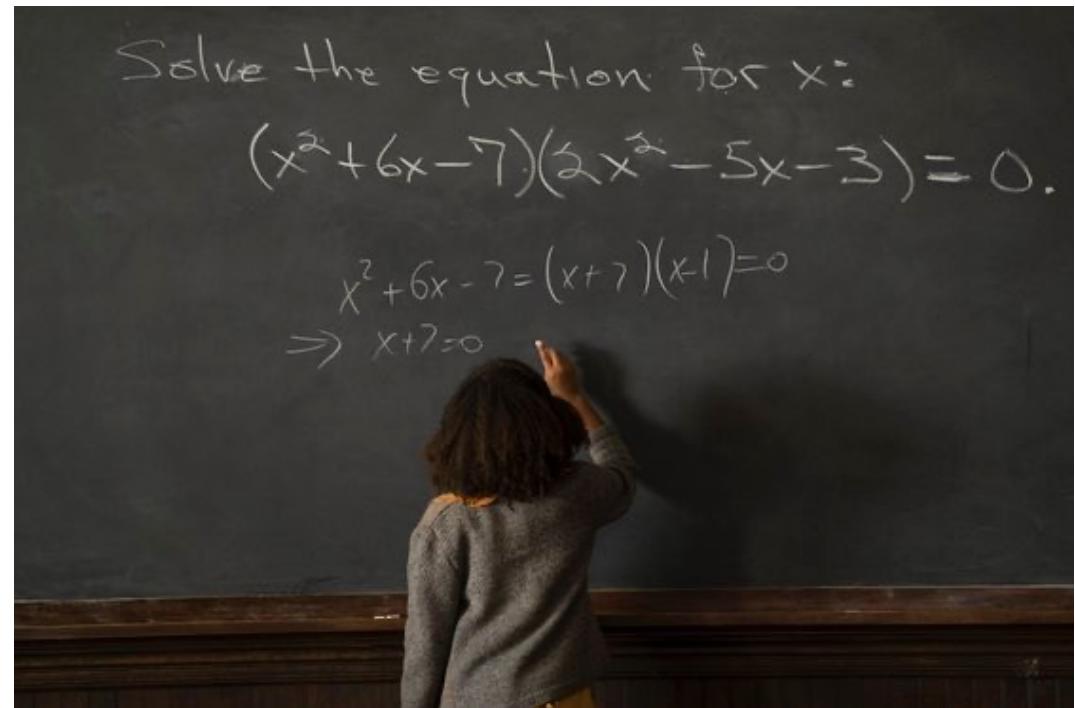
구글 (서울), Core ML model optimization 팀, 소프트웨어 엔지니어 (2022.03 ~ 2022.08)

# 컴퓨터

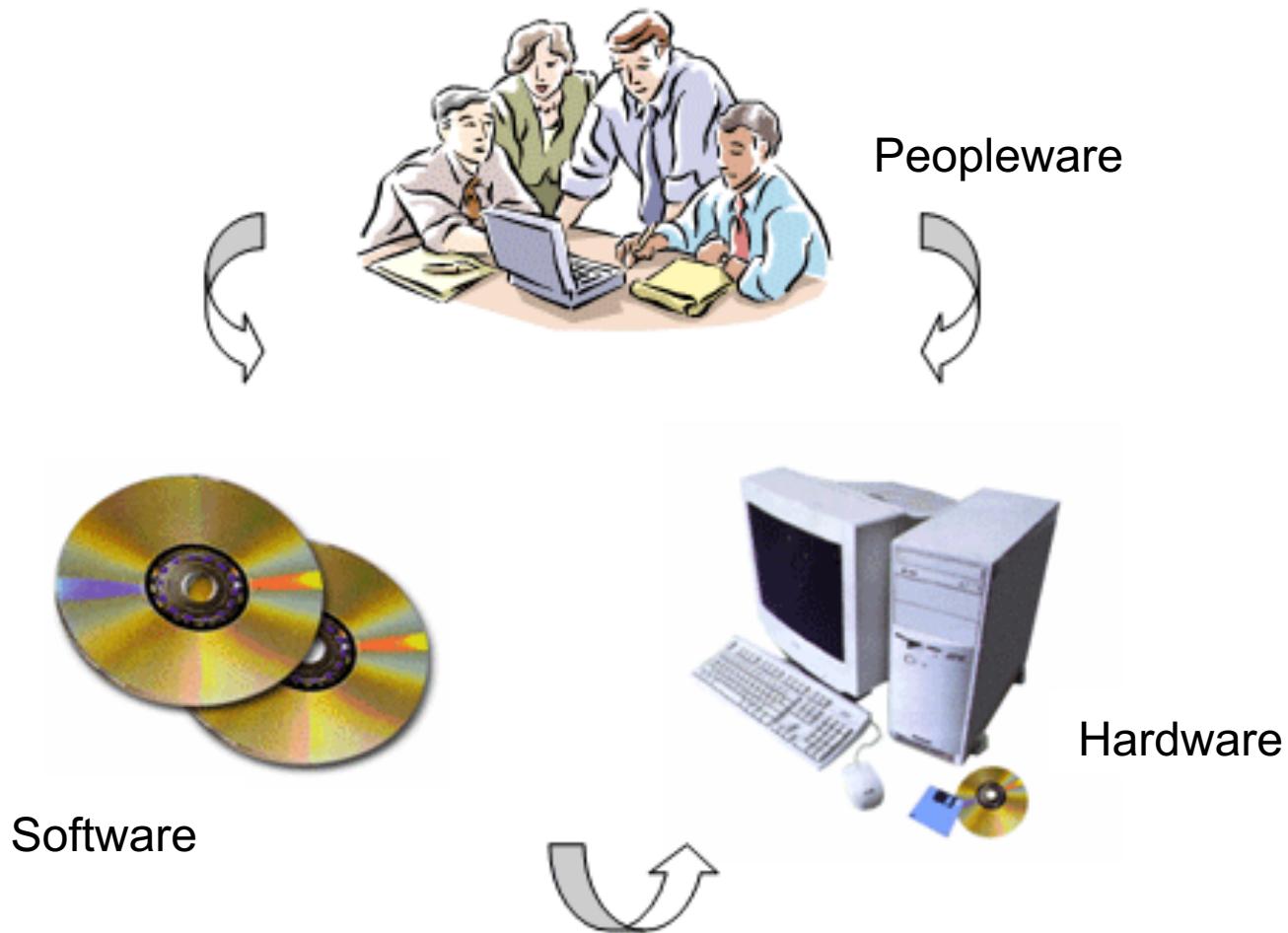
- 컴퓨터의 어원
  - compute + -er
  - '계산자(尺)', '계산수'라는 뜻
  - 뉴턴 이후 물리학에 수학적 기법이 도입되면서 탄도학, 천체물리학 등 물리학 연구에 엄청나게 많은 수학적 계산이 필요
  - 따라서 학자들은 단순 계산을 위한 계산원들을 고용해 쓰기 시작
  - 이 계산원들을 위에서 말한 대로 계산하는 사람이란 뜻의 컴퓨터라고 부른 것이 시초



# 컴퓨터



# 컴퓨터 시스템



From <https://best-knowledge-of-computer.blogspot.com/p/computer-software.html>

# 컴퓨터 시스템

- 현대의 컴퓨터 시스템은 하드웨어 및 소프트웨어로 나뉘어 져 있음



# 컴퓨터 시스템

꽃  
김춘수

내가 그의 이름을 불러주기 전에는  
그는 다만  
하나의 몸짓에 지나지 않았다.

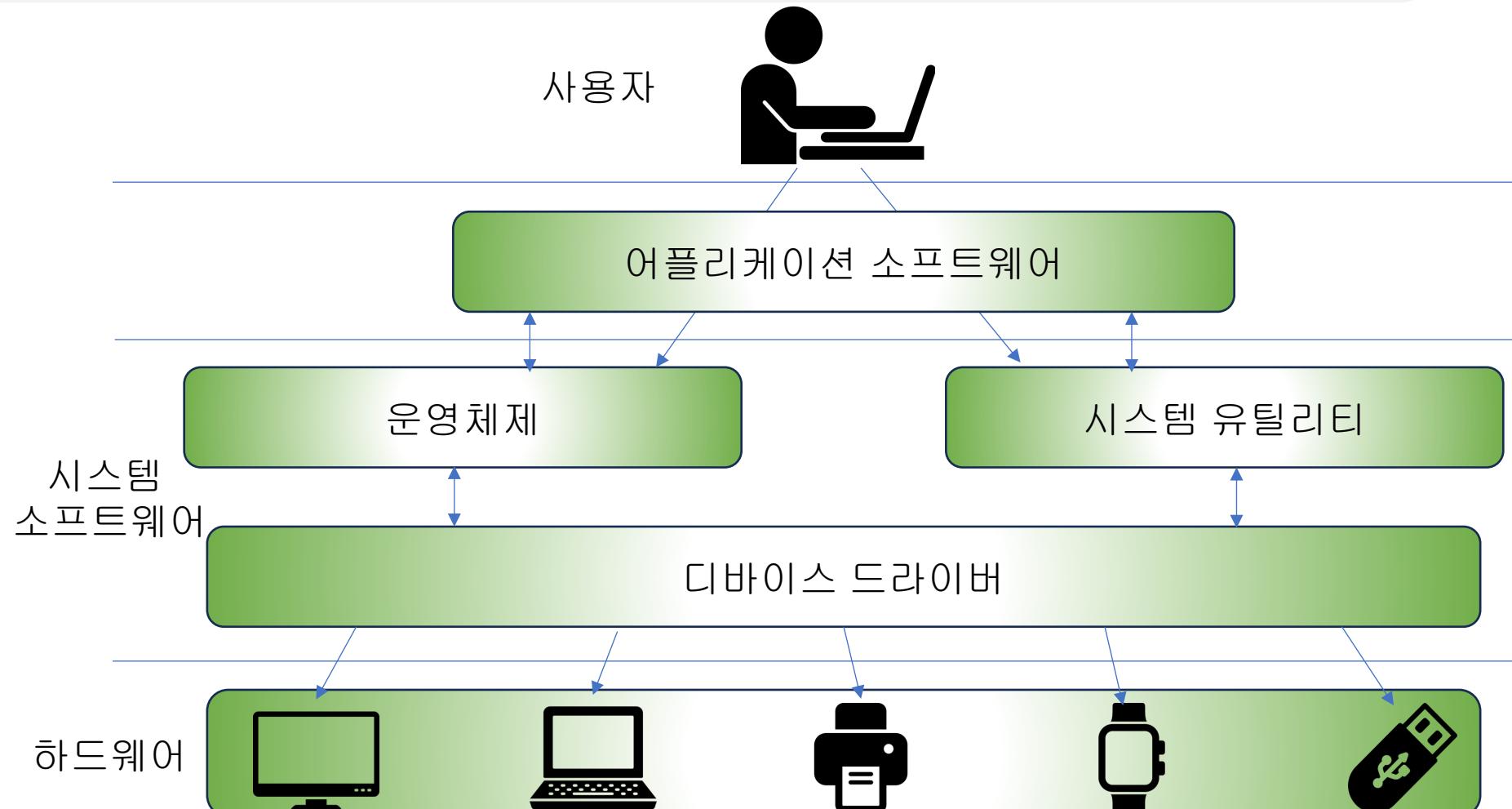
내가 그의 이름을 불러주었을 때,  
그는 나에게로 와서  
꽃이 되었다.

내가 그의 이름을 불러준 것처럼  
나의 이 빛깔과 향기에 알맞는  
누가 나의 이름을 불러다오.  
그에게로 가서 나도  
그의 꽃이 되고 싶다.

우리들은 모두  
무엇이 되고 싶다.  
너는 나에게 나는 너에게  
잊혀지지 않는 하나의 눈짓이 되고 싶다

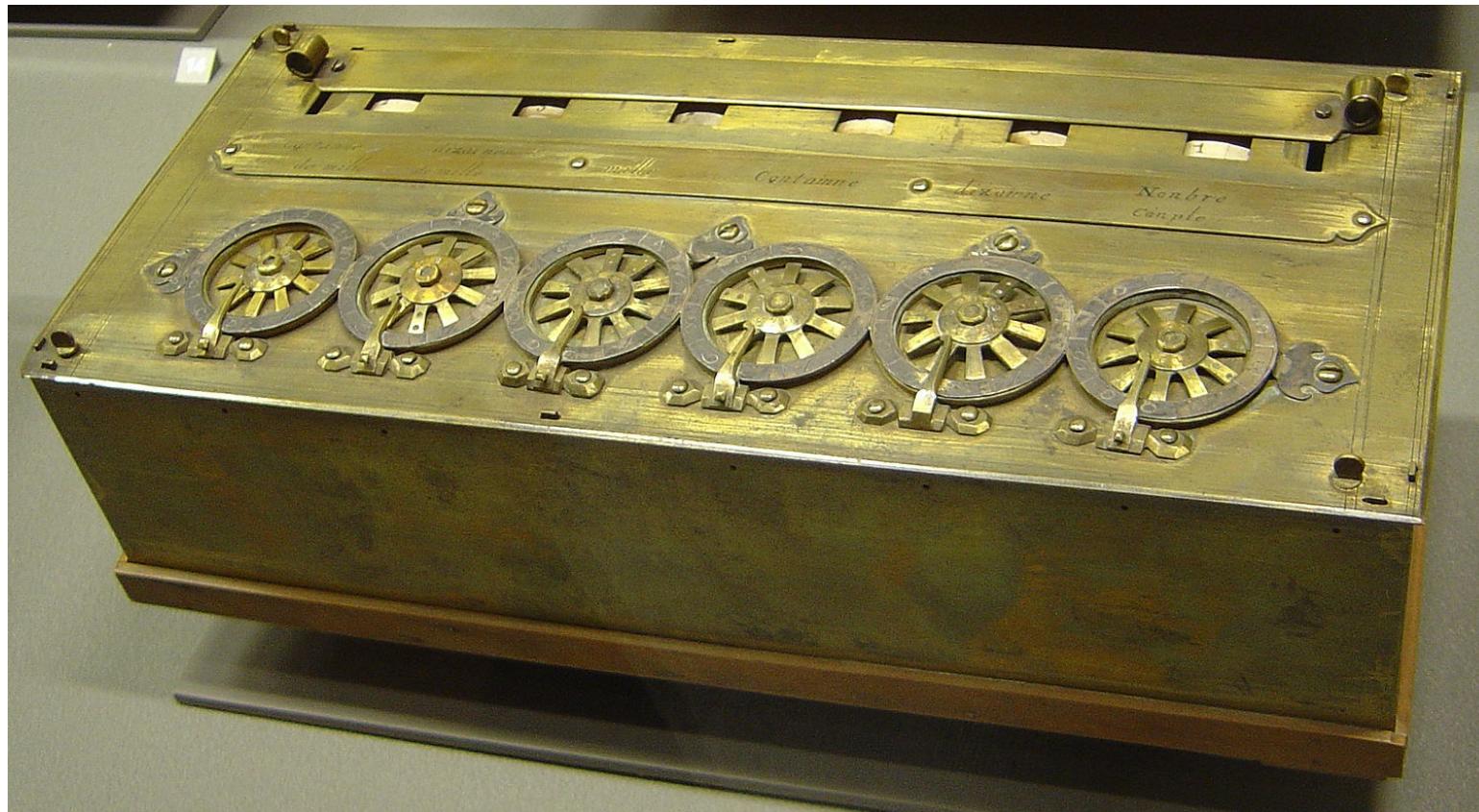


인하대학교  
INHA UNIVERSITY



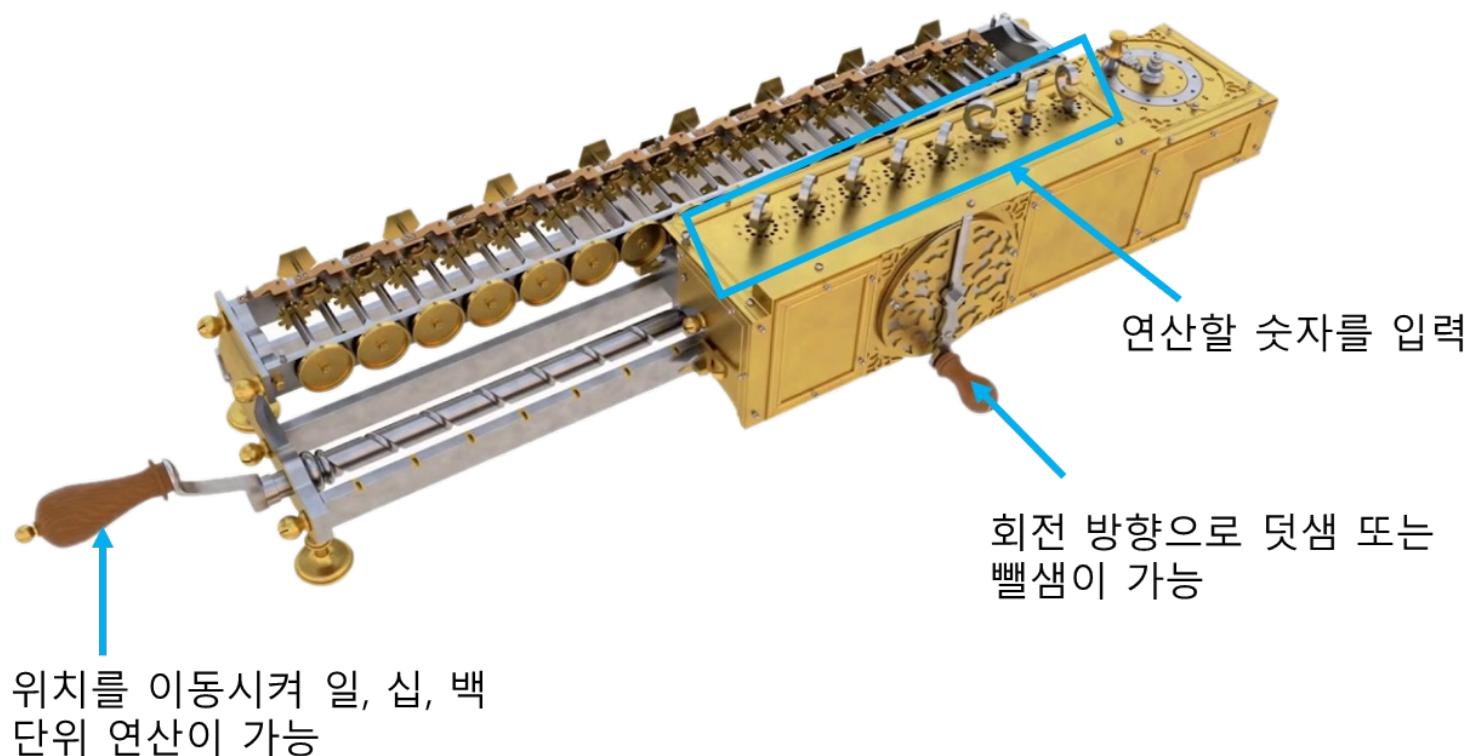
# 컴퓨터 시스템의 역사

- 1642년 파스칼 계산기 – 덧셈, 뺄셈이 가능한 최초의 기계식



# 컴퓨터 시스템의 역사

- 1671년 라이프니츠 계산기 – 곱셈, 나눗셈까지 가능한 사칙연산 계산기

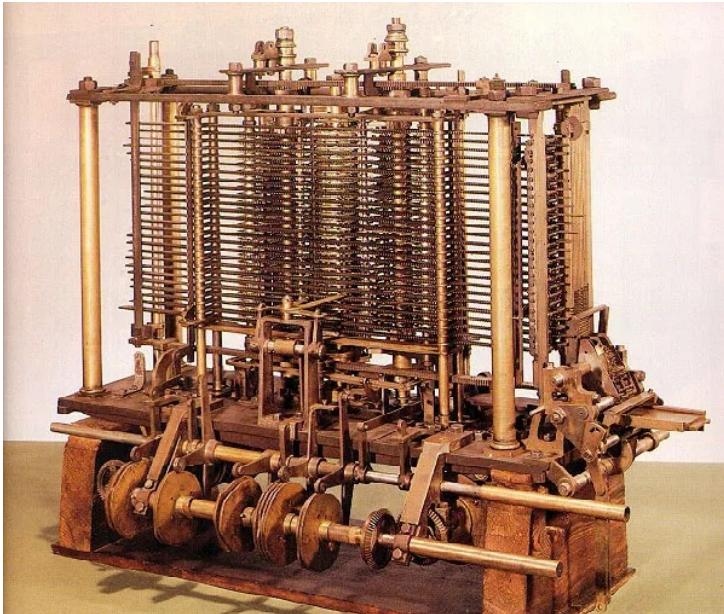


위치를 이동시켜 일, 십, 백  
단위 연산이 가능

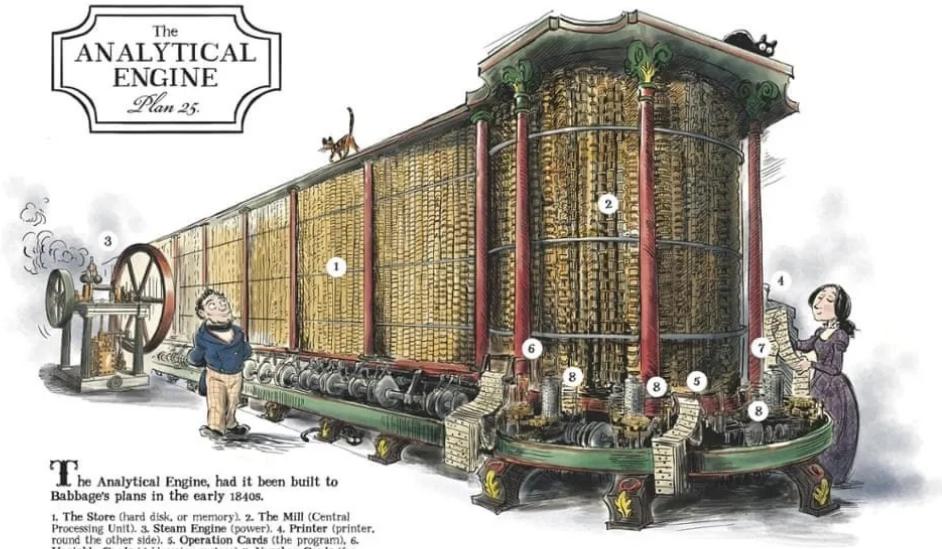
회전 방향으로 덧샘 또는  
뺄샘이 가능

# 컴퓨터 시스템의 역사

- 찰스 배비지 (Charles Babbage ) 의 해석기관 (解析機關, 영어: Analytical Engine) – 현대의 컴퓨터와 매우 흡사한 개념
- 찰스 배비지는 이를 완성하지 못하고 1871년 세상을 뜨게 됨



미완성 모형



완성 된 해석기관의 상상도

From [https://computerscience.fandom.com/wiki/History\\_of\\_Computers](https://computerscience.fandom.com/wiki/History_of_Computers)

From <https://www.theguardian.com/technology/2015/apr/12/thrilling-adventures-ada-lovelace-charles-babbage-sydney-padua>

# 컴퓨터 시스템의 역사

- 에이다 러브레이스 (1815~1852) – 인류 역사상 최초의 프로그래머
  - 만들어지지도 않은 해석기관을 이용해 베르누이 수를 구하는 알고리즘을 작성
  - 현재 알려진 최초의 컴퓨터 프로그램

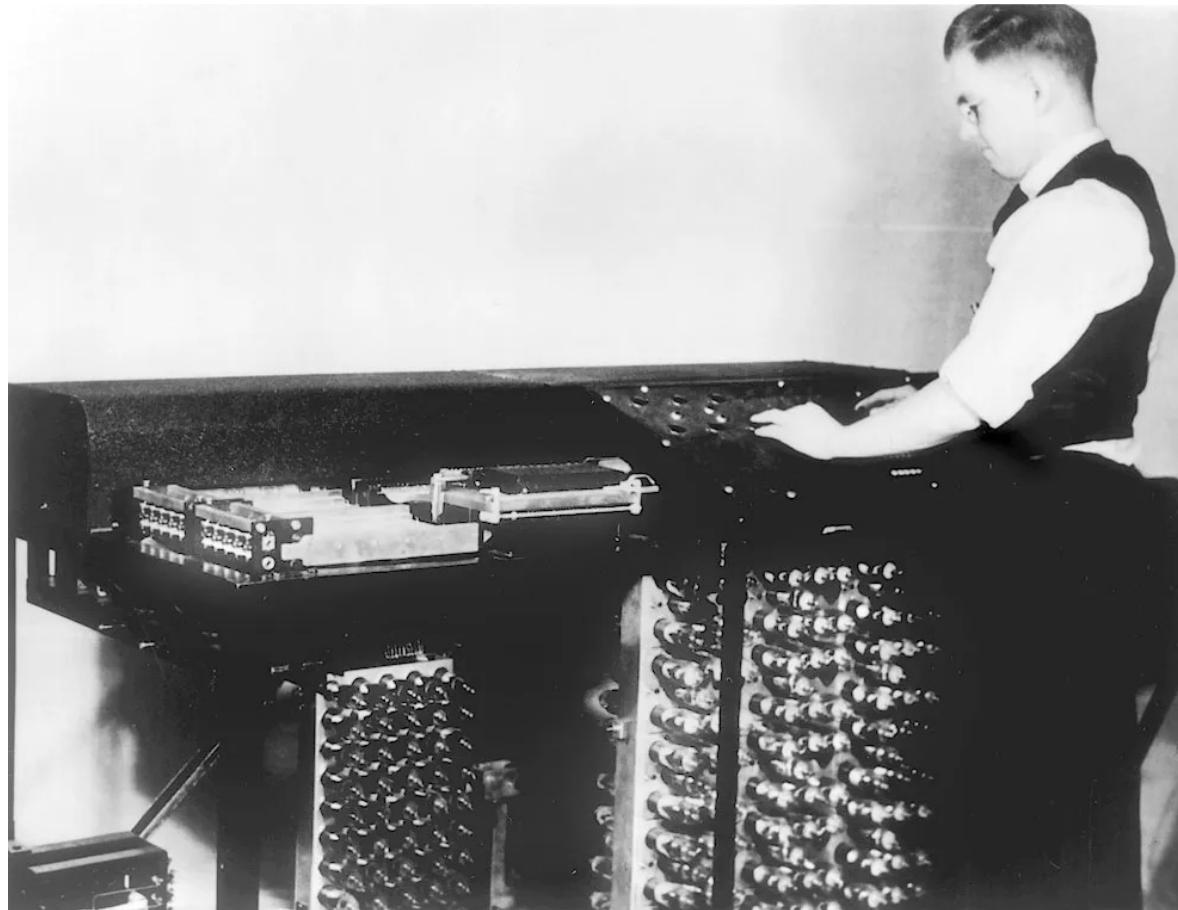


Number of Operation. Nature of Operation.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data.												Working Variables.												Result Variables.	
					v <sub>1</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>5</sub>	v <sub>6</sub>	v <sub>7</sub>	v <sub>8</sub>	v <sub>9</sub>	v <sub>10</sub>	v <sub>11</sub>	v <sub>12</sub>	v <sub>13</sub>	v <sub>14</sub>	v <sub>15</sub>	v <sub>16</sub>	v <sub>17</sub>	v <sub>18</sub>	v <sub>19</sub>	v <sub>20</sub>	v <sub>21</sub>	v <sub>22</sub>	v <sub>23</sub>	v <sub>24</sub>	v <sub>25</sub>	v <sub>26</sub>
1	$\times$	$v_2 \times v_3$	$v_{v_4}, v_{v_5}, v_{v_6}$	$\begin{cases} v_{v_5} = v_{v_5} \\ v_{v_5} = v_{v_5} \end{cases}$	$= 2n$																									
2	$-$	$v_4 - v_1$	$v_{v_4}$	$\begin{cases} v_{v_4} = v_{v_4} \\ v_{v_4} = v_{v_4} \end{cases}$	$= 2n - 1$																									
3	$+$	$v_3 + v_1$	$v_{v_3}$	$\begin{cases} v_{v_3} = v_{v_3} \\ v_{v_3} = v_{v_3} \end{cases}$	$= 2n + 1$																									
4	$+$	$v_5 + v_4$	$v_{v_11}$	$\begin{cases} v_{v_5} = v_{v_5} \\ v_{v_5} = v_{v_5} \end{cases}$	$= 2n + 1$																									
5	$+$	$v_{11} + v_2$	$v_{v_11}$	$\begin{cases} v_{v_11} = v_{v_11} \\ v_{v_11} = v_{v_11} \end{cases}$	$= \frac{1}{2} \cdot 2n - 1$																									
6	$-$	$v_{13} - v_{11}$	$v_{v_{12}}$	$\begin{cases} v_{v_{13}} = v_{v_{13}} \\ v_{v_{13}} = v_{v_{13}} \end{cases}$	$= -\frac{1}{2} \cdot 2n + 1 = A_0$																									
7	$-$	$v_3 - v_1$	$v_{v_{10}}$	$\begin{cases} v_{v_3} = v_{v_3} \\ v_1 = v_1 \end{cases}$	$= n - 1 = (3)$																									
8	$+$	$v_2 + v_7$	$v_{v_2}$	$\begin{cases} v_{v_2} = v_{v_2} \\ v_{v_2} = v_{v_2} \end{cases}$	$= 2 + 0 = 2$																									
9	$+$	$v_6 + v_7$	$v_{v_{11}}$	$\begin{cases} v_{v_6} = v_{v_6} \\ v_7 = v_7 \end{cases}$	$= \frac{n}{2} = A_1$																									
10	$\times$	$v_{21} \times v_{11}$	$v_{v_{12}}$	$\begin{cases} v_{v_{21}} = v_{v_{21}} \\ v_{v_{11}} = v_{v_{11}} \end{cases}$	$= B_1 \cdot \frac{2n}{2} = B_1 A_1$																									
11	$+$	$v_{12} + v_{13}$	$v_{v_{12}}$	$\begin{cases} v_{v_{12}} = v_{v_{12}} \\ v_{v_{13}} = v_{v_{13}} \end{cases}$	$= -\frac{1}{2} \cdot 2n - 1 + B_1 \cdot \frac{2n}{2}$																									
12	$-$	$v_{10} - v_1$	$v_{v_{10}}$	$\begin{cases} v_{v_{10}} = v_{v_{10}} \\ v_1 = v_1 \end{cases}$	$= n - 2 = (2)$																									
13	$-$	$v_6 - v_1$	$v_{v_6}$	$\begin{cases} v_{v_6} = v_{v_6} \\ v_1 = v_1 \end{cases}$	$= 2n - 1$																									
14	$+$	$v_1 + v_2$	$v_{v_7}$	$\begin{cases} v_1 = v_1 \\ v_2 = v_2 \end{cases}$	$= 2 + 1 = 3$																									
15	$+$	$v_6 + v_7$	$v_{v_8}$	$\begin{cases} v_{v_6} = v_{v_6} \\ v_7 = v_7 \end{cases}$	$= \frac{2n}{3} - 1$																									
16	$\times$	$v_8 \times v_{11}$	$v_{v_{11}}$	$\begin{cases} v_{v_8} = v_{v_8} \\ v_{v_{11}} = v_{v_{11}} \end{cases}$	$= \frac{2n}{3} \cdot \frac{2n - 1}{3} = B_1 A_1$																									
17	$-$	$v_6 - v_1$	$v_{v_6}$	$\begin{cases} v_{v_6} = v_{v_6} \\ v_1 = v_1 \end{cases}$	$= 2n - 2$																									
18	$+$	$v_5 + v_7$	$v_{v_7}$	$\begin{cases} v_{v_5} = v_{v_5} \\ v_7 = v_7 \end{cases}$	$= 3 + 1 = 4$																									
19	$+$	$v_6 + v_7$	$v_{v_9}$	$\begin{cases} v_{v_6} = v_{v_6} \\ v_7 = v_7 \end{cases}$	$= \frac{2n}{3} - 2$																									
20	$\times$	$v_9 \times v_{11}$	$v_{v_{11}}$	$\begin{cases} v_{v_9} = v_{v_9} \\ v_{v_{11}} = v_{v_{11}} \end{cases}$	$= B_2 \cdot \frac{2n}{3} \cdot \frac{2n - 1}{3} \cdot \frac{2n - 2}{3} = A_3$																									
21	$\times$	$v_{10} \times v_{12}$	$v_{v_{12}}$	$\begin{cases} v_{v_{10}} = v_{v_{10}} \\ v_{v_{12}} = v_{v_{12}} \end{cases}$	$= B_2 \cdot \frac{2n}{3} \cdot \frac{2n - 1}{3} \cdot \frac{2n - 2}{3} \cdot R_2 A_2 = 0$																									
22	$+$	$v_5 + v_{12}$	$v_{v_{12}}$	$\begin{cases} v_{v_5} = v_{v_5} \\ v_{v_{12}} = v_{v_{12}} \end{cases}$	$= A_0 + B_1 A_1 + B_2 A_2$																									
23	$-$	$v_{10} - v_1$	$v_{v_{10}}$	$\begin{cases} v_{v_{10}} = v_{v_{10}} \\ v_1 = v_1 \end{cases}$	$= n - 3 = (1)$																									
24	$+$	$v_{13} + v_{20}$	$v_{v_{24}}$	$\begin{cases} v_{v_{13}} = v_{v_{13}} \\ v_{v_{20}} = v_{v_{20}} \end{cases}$	$= B_7$																									
25	$+$	$v_1 + v_5$	$v_{v_3}$	$\begin{cases} v_1 = v_1 \\ v_5 = v_5 \end{cases}$	$= n + 1 = 4 + 1 = 5$																									

Here follows a repetition of Operations thirteen to twenty-three.

# 컴퓨터 시스템의 역사

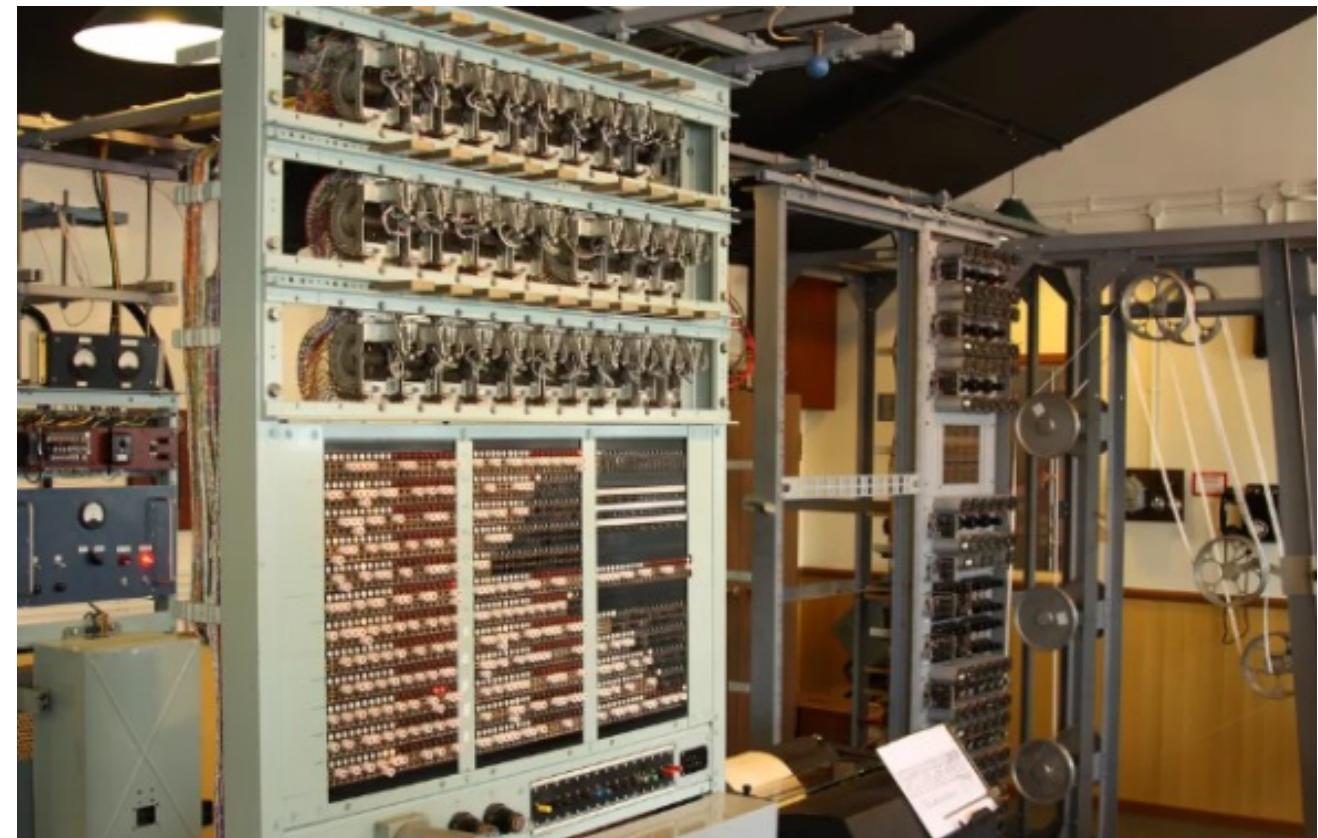
- 1942년 아타나소프-베리 컴퓨터(Atanasoff-Berry Computer) – 세계 최초의 전자식 컴퓨터



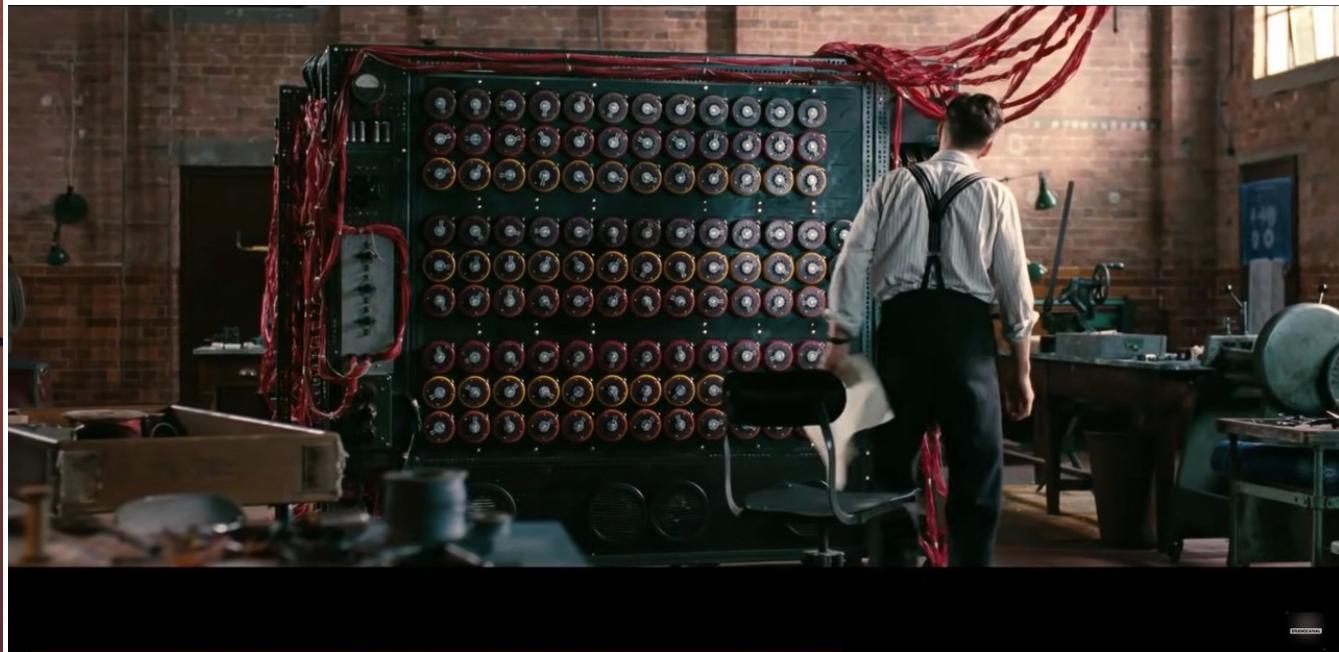
From <https://www.pinterest.com/pin/136163588708805707/>

# 컴퓨터 시스템의 역사

- 1943 앤런 튜링의 콜로서스(Colossus) – 프로그래밍 가능 디지털 전자 컴퓨터



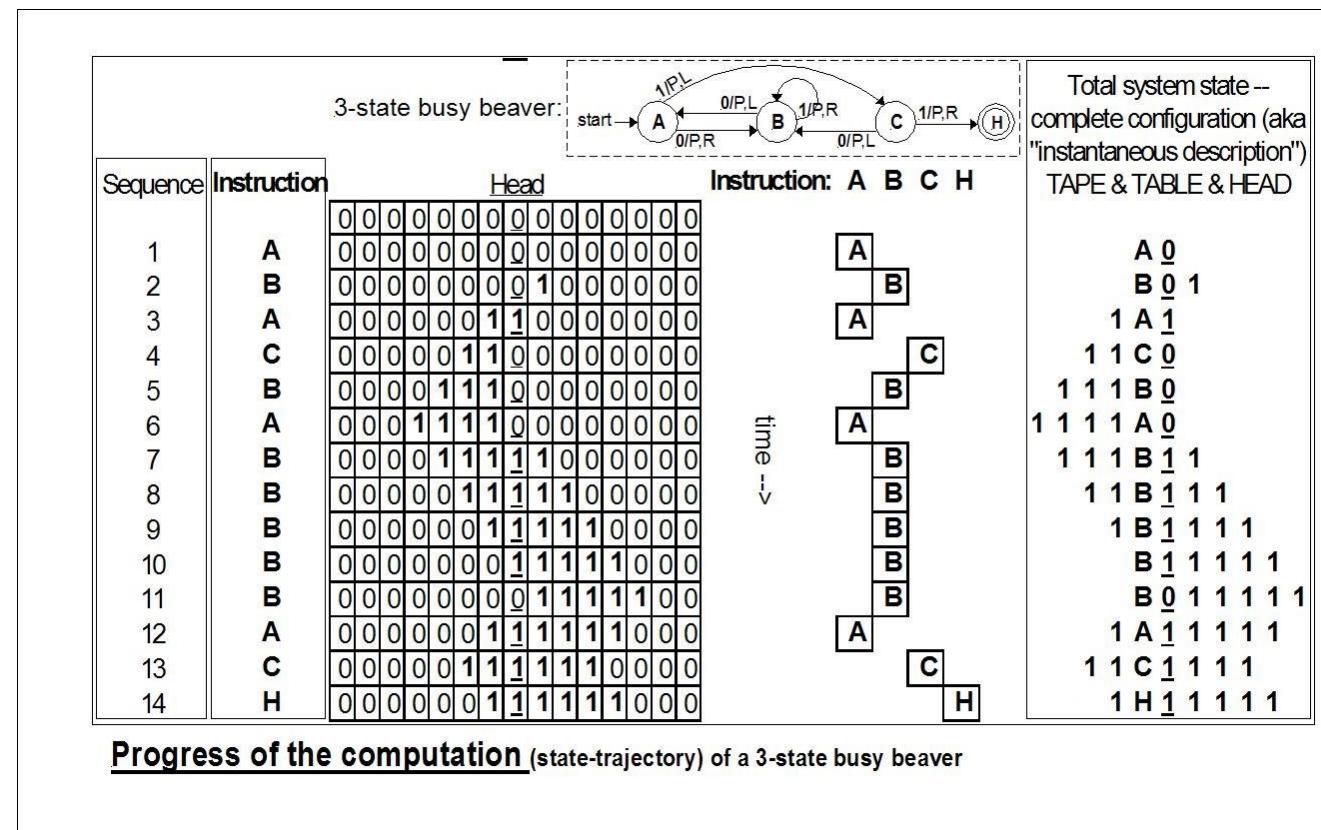
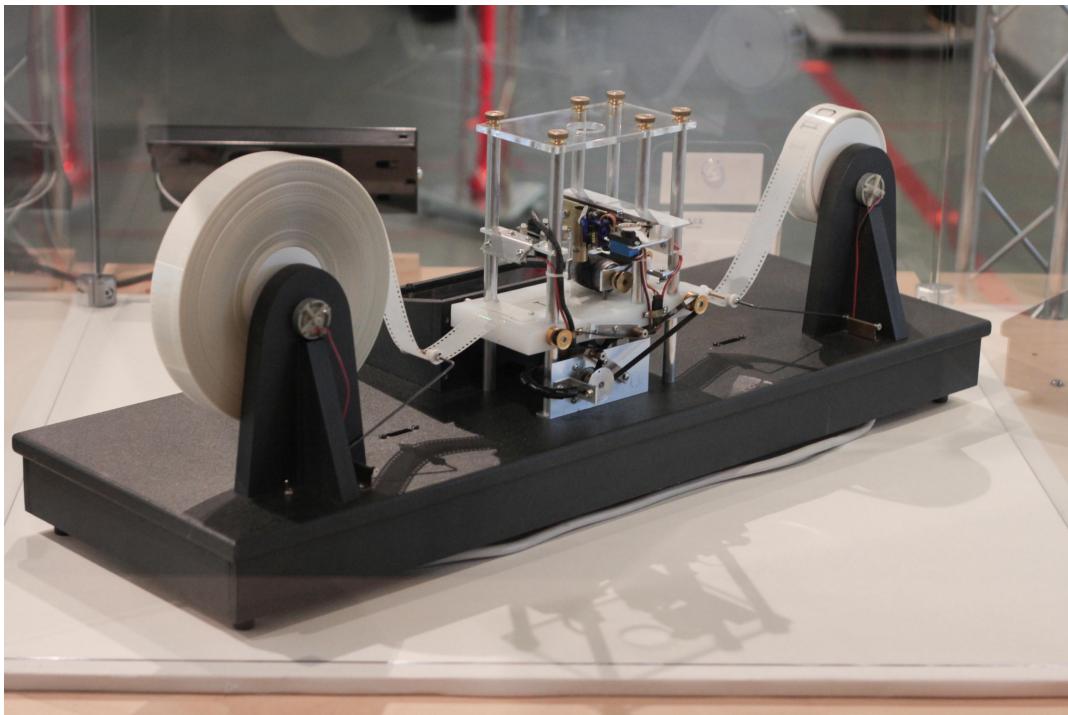
# 컴퓨터 시스템의 역사



인하대학교  
INHA UNIVERSITY

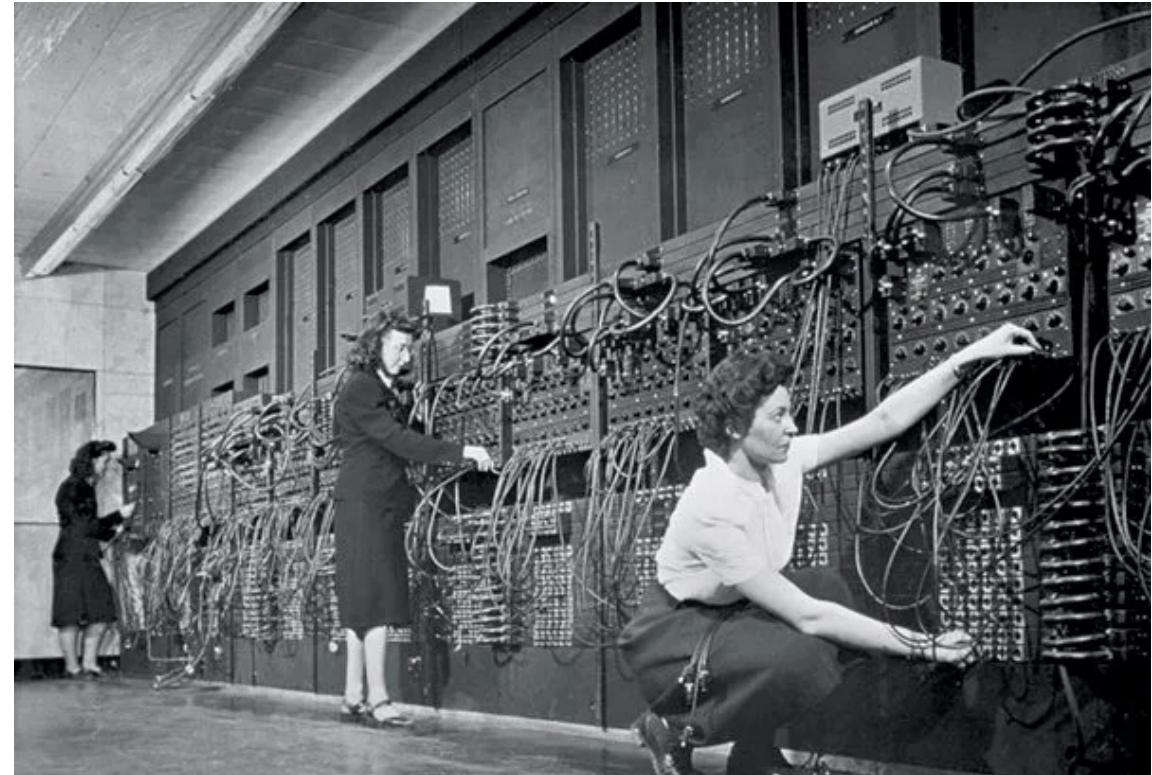
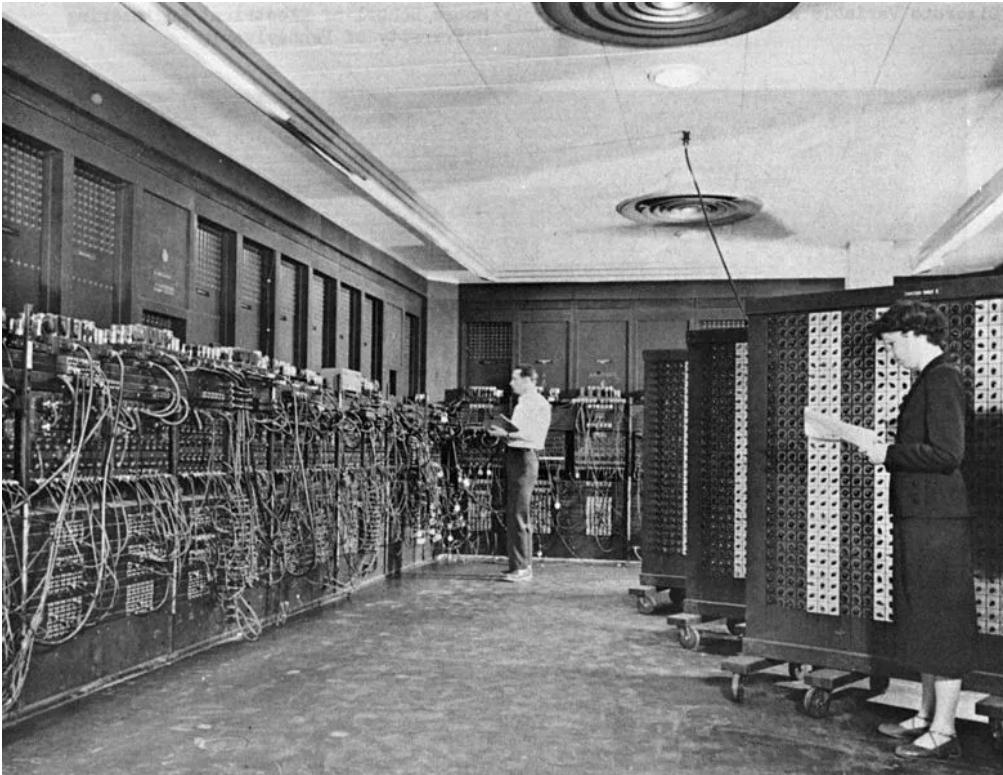
# 컴퓨터 시스템의 역사

- 튜링 머신 – 앤런 튜링이 1936년에 제시한 가상의 기계
- 튜링 완전 언어 – 언어나 시스템이 튜링 머신과 동일한 계산 능력을 가지고 있음을 의미이며, 현대 프로그래밍 언어의 근원



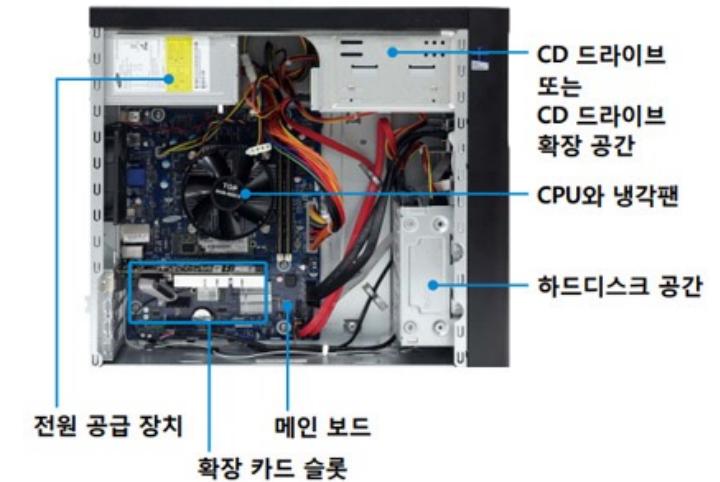
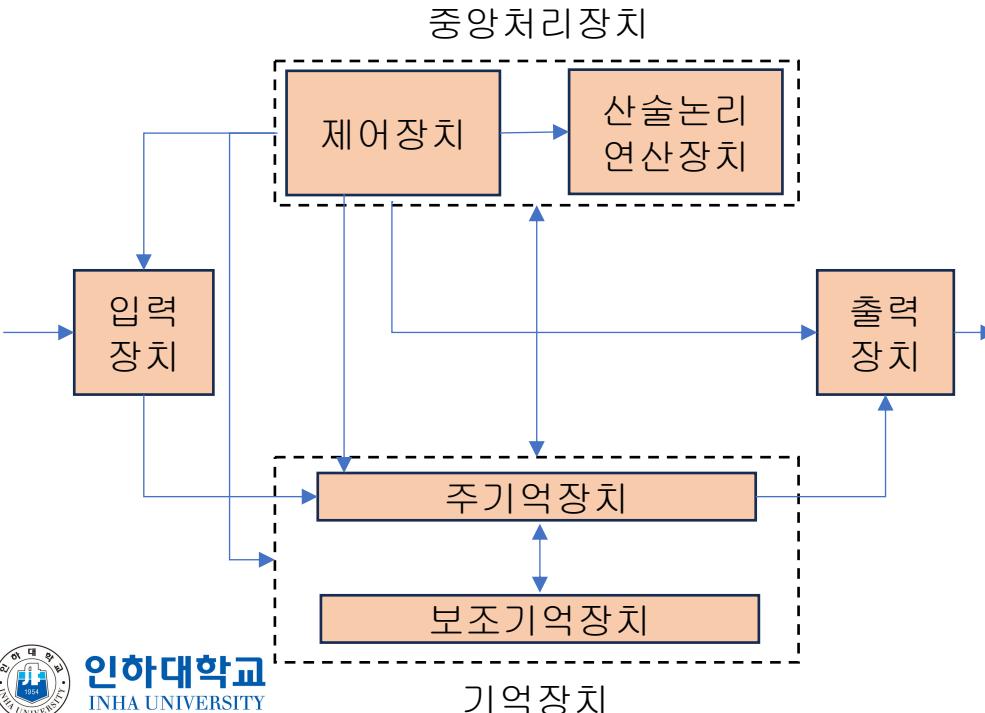
# 컴퓨터 시스템의 역사

- 1946년 폰 노이만의 애니악 (ENIAC – Electronic Numerical Integrator And Calculator)

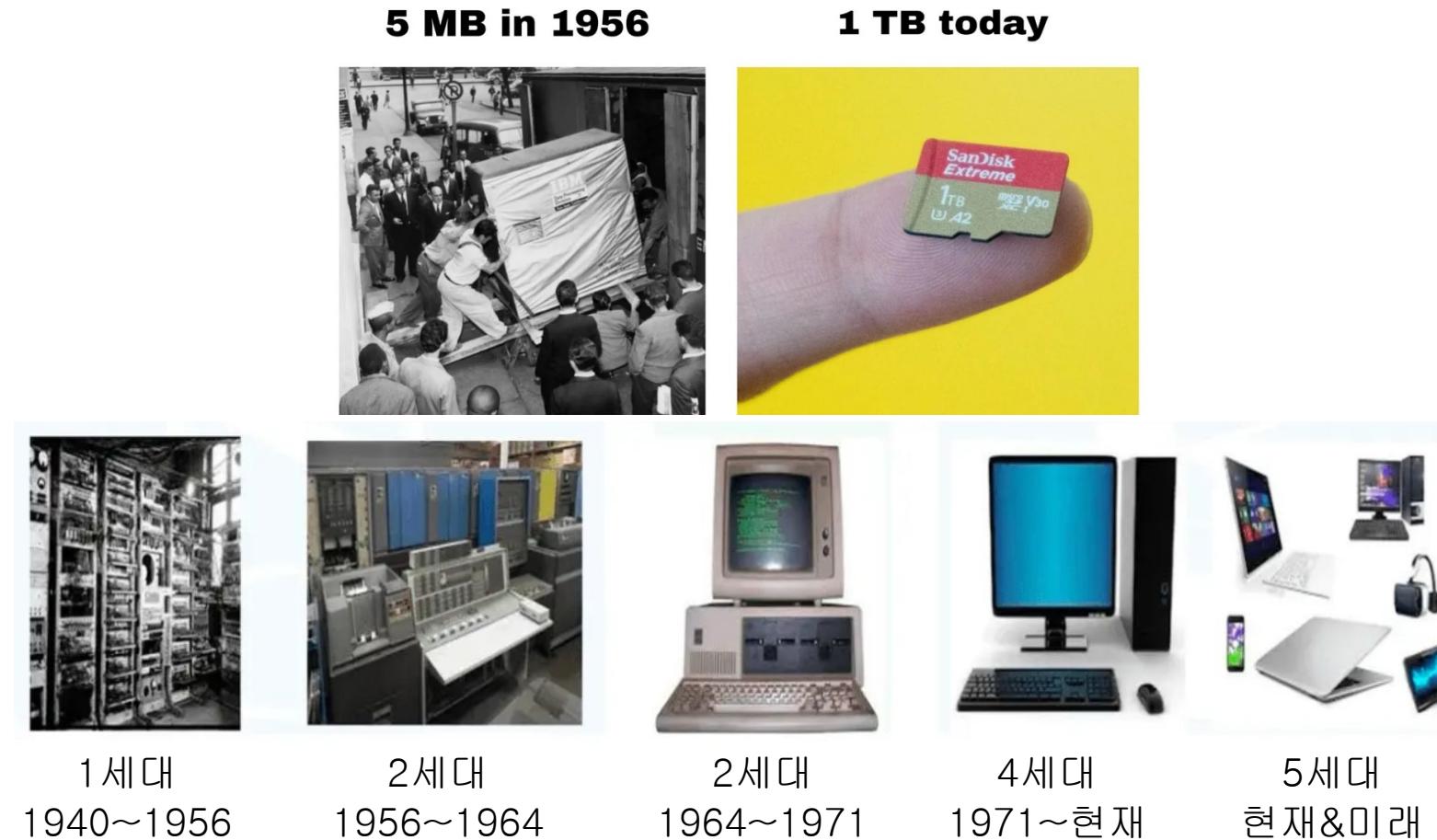
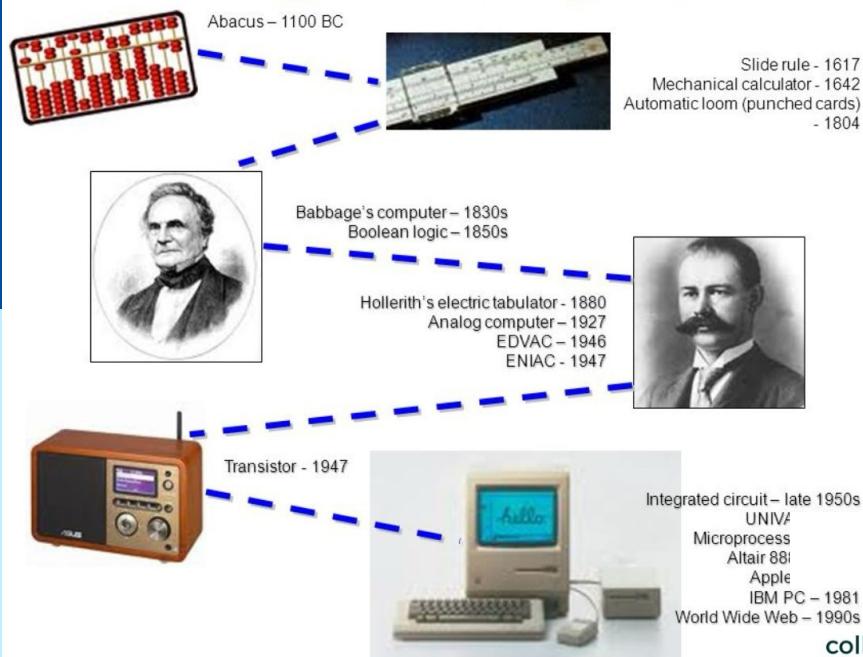


# 컴퓨터 시스템의 역사

- 폰 노이만 구조
  - 수학자이자 물리학자 폰 노이만과 다른 사람들이 1945년 제안한 컴퓨터 구조
  - 맨해튼 프로젝트에 참여할 당시 발표한 논문 <전자계산기의 이론 설계 서론>에서 CPU, 메모리, 프로그램 구조를 갖는 프로그램 내장 방식 컴퓨터의 아이디어를 처음 제시



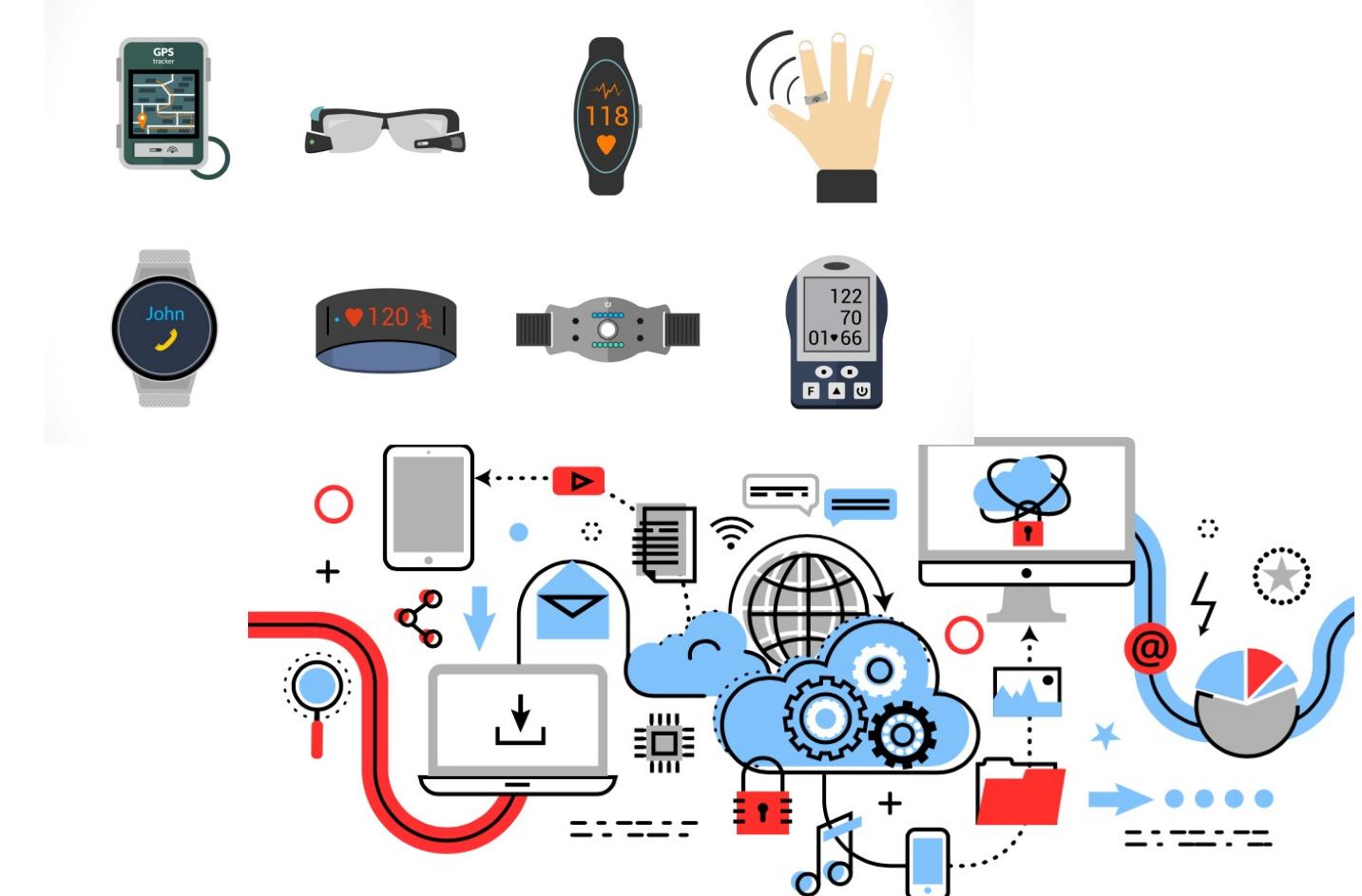
# 컴퓨터 시스템의 역사



From <https://www.collegesearch.in/articles/history-of-computer>

From [https://www.reddit.com/r/pcmasterrace/comments/edvwvu/look\\_at\\_how\\_far\\_weve\\_come/](https://www.reddit.com/r/pcmasterrace/comments/edvwvu/look_at_how_far_weve_come/)

# 컴퓨터 시스템의 역사

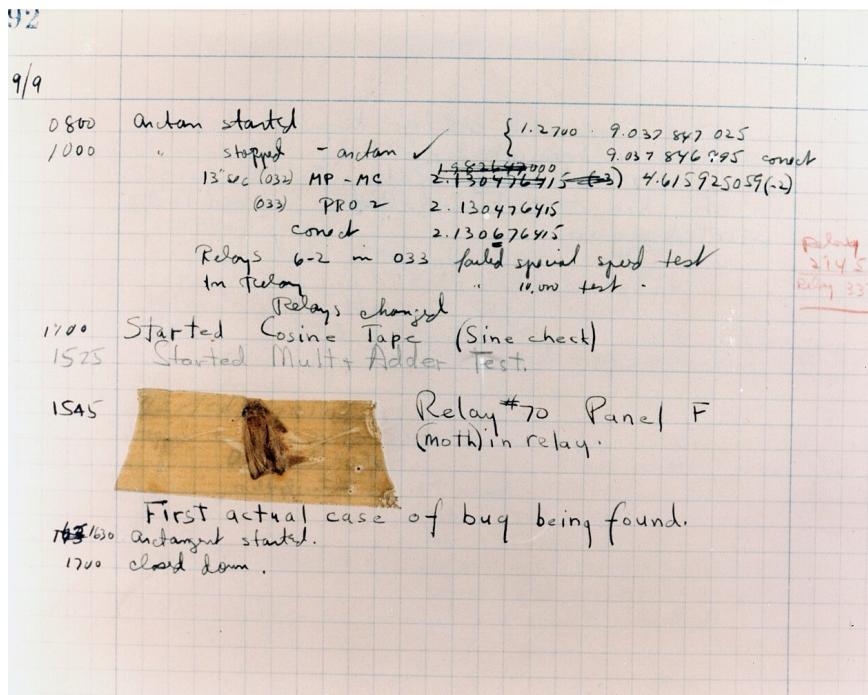
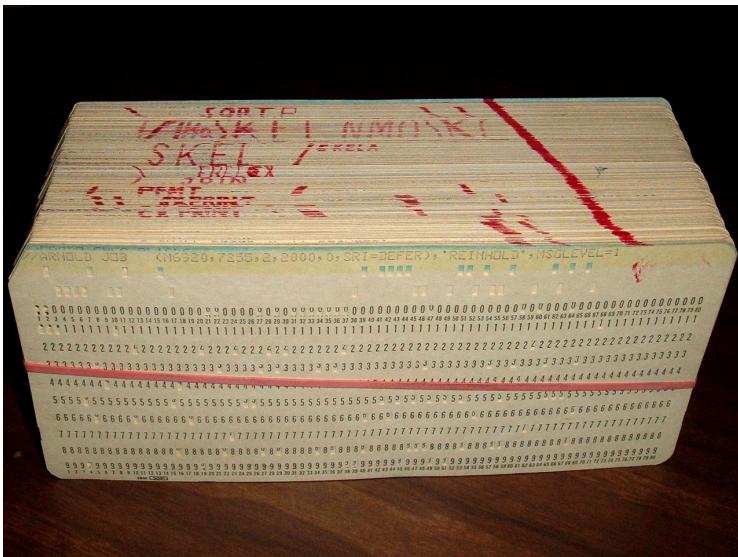
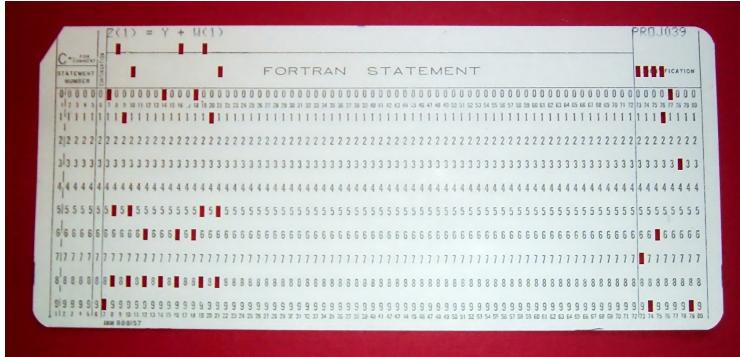


From [https://www.123rf.com/photo\\_20391699\\_computer-hardware-network-and-mobile-devices-set.html](https://www.123rf.com/photo_20391699_computer-hardware-network-and-mobile-devices-set.html)

From <https://techcrunch.com/2015/06/10/a-periodic-table-of-wearable-technology/>

From <https://clovers.law/en/blog/2020/6/1/6lu6lf2soto2fhelekwtuykrmc5891>

# 컴퓨터 시스템의 역사



```

MONITOR FOR 6802 1.4          9-14-80  TSC ASSEMBLER PAGE 2

C000 C000 8E 00 70 START ORG ROM+$0000 BEGIN MONITOR
LDS #STACK

*****+
* FUNCTION: INITA - Initialize ACIA
* INPUT: none
* OUTPUT: none
* CALLS: none
* DESTROYS: acc A
0013 RESETA EQU %00010011
0011 CTLREG EQU %00010001

C003 86 13 INITA LDA A #RESETA RESET ACIA
C005 B7 80 04 STA A ACIA
C008 86 11 LDA A #CTLREG SET 8 BITS AND 2 STOP
C00A B7 80 04 STA A ACIA
C00D 7E C0 F1 JMP SIGNON GO TO START OF MONITOR

*****+
* FUNCTION: INCH - Input character
* INPUT: none
* OUTPUT: char in acc A
* DESTROYS: acc A
* CALLS: none
* DESCRIPTION: Gets 1 character from terminal
C010 B6 80 04 INCH LDA A ACIA GET STATUS
C013 47 ASK A BCC INCH SHIFT RDRF FLAG INTO CARRY
C014 24 FA RECIEVE NOT READY
C016 B6 80 05 LDA A ACIA+1 GET CHAR
C019 84 7F AND A #57F MASK PARITY
C01B 7E C0 79 JMP OUTCH ECHO & RTS

*****+
* FUNCTION: INHEX - INPUT HEX DIGIT
* INPUT: none
* OUTPUT: Digit in acc A
* CALLS: INCH
* DESTROYS: acc A
* Returns to monitor if not HEX input
C01E 8D F0 INHEX BSR INCH GET A CHAR
C020 81 30 CMP A #'0 ZERO
C022 2B 11 BMI HEXERR NOT HEX
C024 81 39 CMP A #'9 NINE
C026 2F 0A BLE HEXRTS GOOD HEX
C028 81 41 CMP A #'A
C02A 2B 09 BMI HEXERR
C02C 81 46 CMP A #'F
C02E 2E 05 BGT HEXERR
C030 80 07 SUB A #7 FIX A-F
C032 84 0F HEXRTS AND A #50F CONVERT ASCII TO DIGIT
C034 39 RTS
C035 7E C0 AF HEXERR JMP CTRL RETURN TO CONTROL LOOP

```

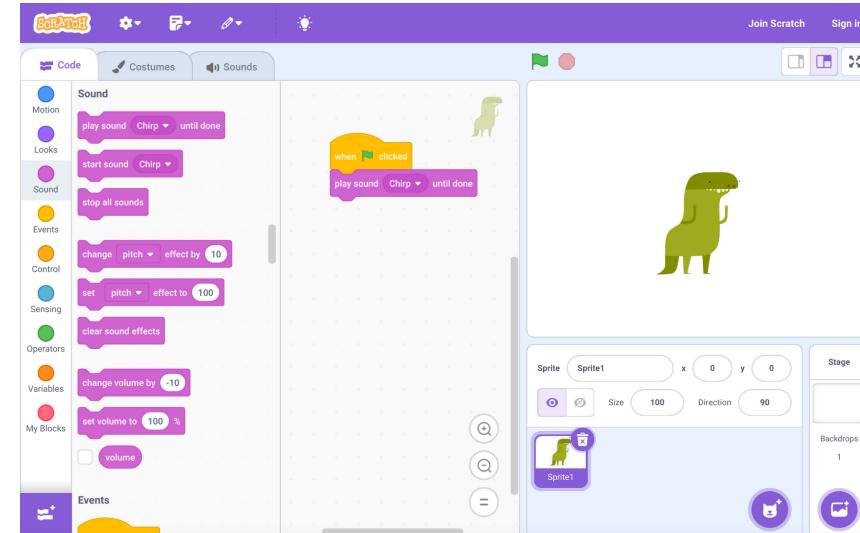
From [https://en.wikipedia.org/wiki/Computer\\_programming\\_in\\_the\\_punched\\_card\\_era](https://en.wikipedia.org/wiki/Computer_programming_in_the_punched_card_era)  
 From [https://en.wikipedia.org/wiki/Computer\\_program](https://en.wikipedia.org/wiki/Computer_program)  
 From [https://en.wikipedia.org/wiki/Assembly\\_language](https://en.wikipedia.org/wiki/Assembly_language)

# 컴퓨터 시스템의 역사

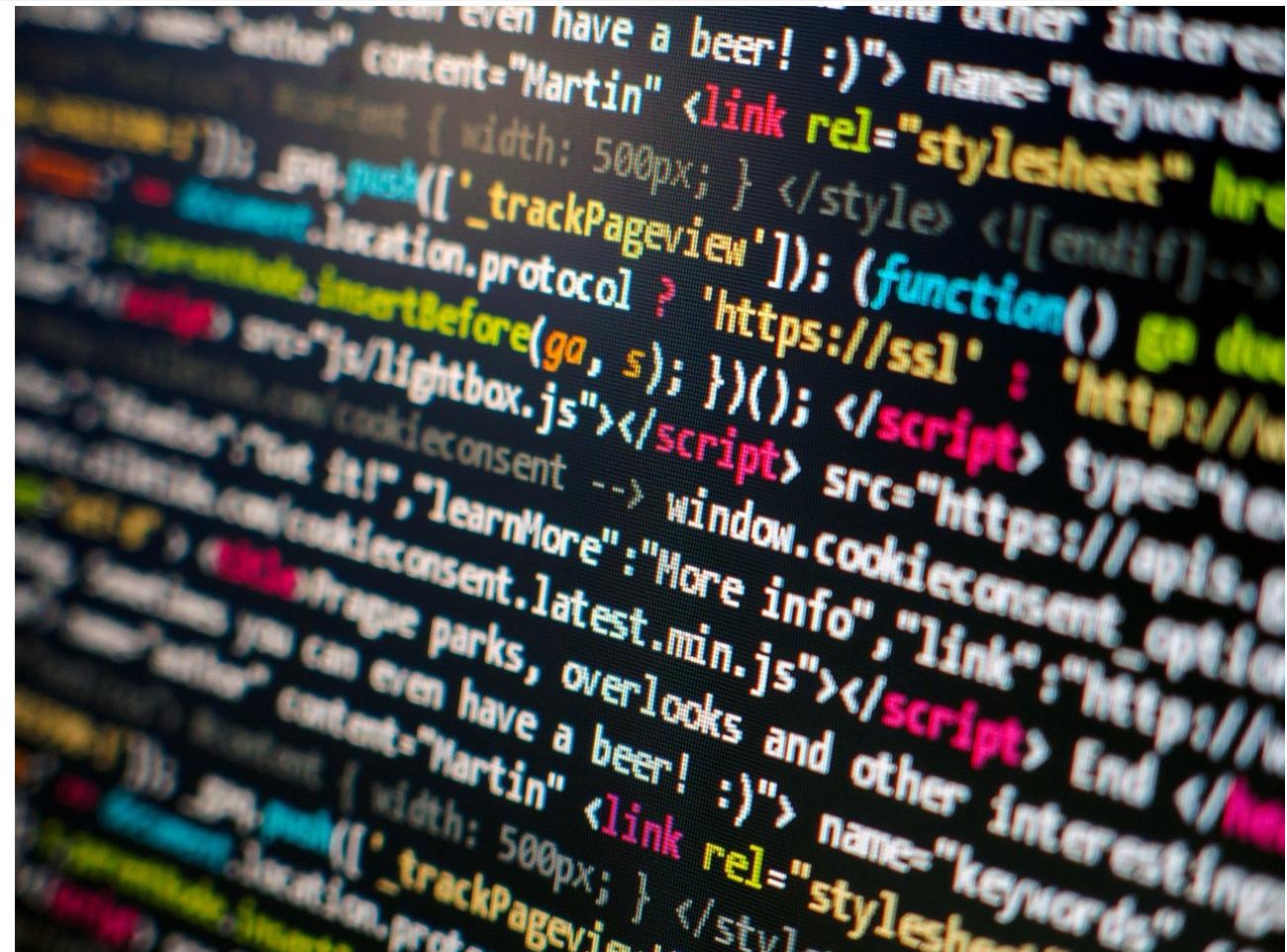
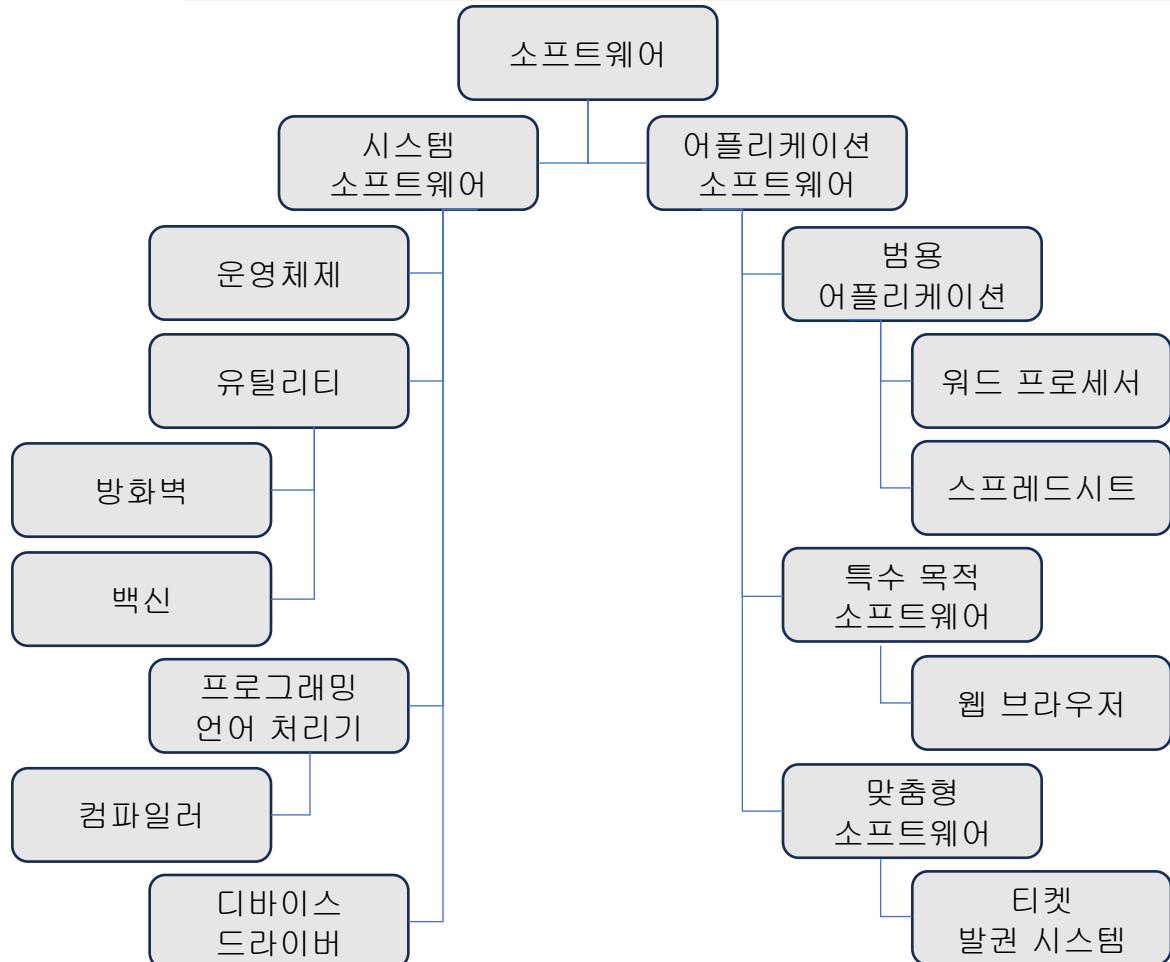
```
1  sub Calculator()
2    sub addition(self, other)
3      return self + other
4    end
5
6    /** Create a list of 2 numbers. */
7    sub makeFraction(numerator, denominator)
8      return [numerator, denominator]
9    end
10
11   /** Warning: Destroys original fraction! */
12   sub multiplyFracs(frac, otherFrac)
13     frac[0] *= otherFrac[0]
14     frac[1] *= otherFrac[1]
15     return frac
16   end
17
18   sub InfinityCalculator()
19     inherit Calculator()
20     /** Create a list of 2 numbers. */
21     sub makeFraction(numerator, denominator)
22       if denominator == 0
23         /* The user is trying to divide by 0.
24         * Use Java's way of handling this: */
25         import math into mathematics
26         return mathematics.INFINITY
27       end
28       return [numerator, denominator]
29     end
30   end
31 end
```

Method

Class



# 컴퓨터 시스템의 역사



From [https://en.wikipedia.org/wiki/Computer\\_program](https://en.wikipedia.org/wiki/Computer_program)



# 미래의 컴퓨터 시스템



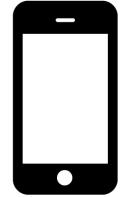
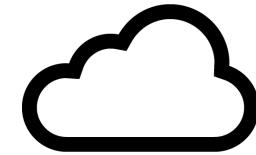
From [http://www.impawards.com/2015/pixels\\_ver6\\_xlg.html](http://www.impawards.com/2015/pixels_ver6_xlg.html)

From <https://www.linkedin.com/pulse/software-eating-world-peter-balz-1e/>

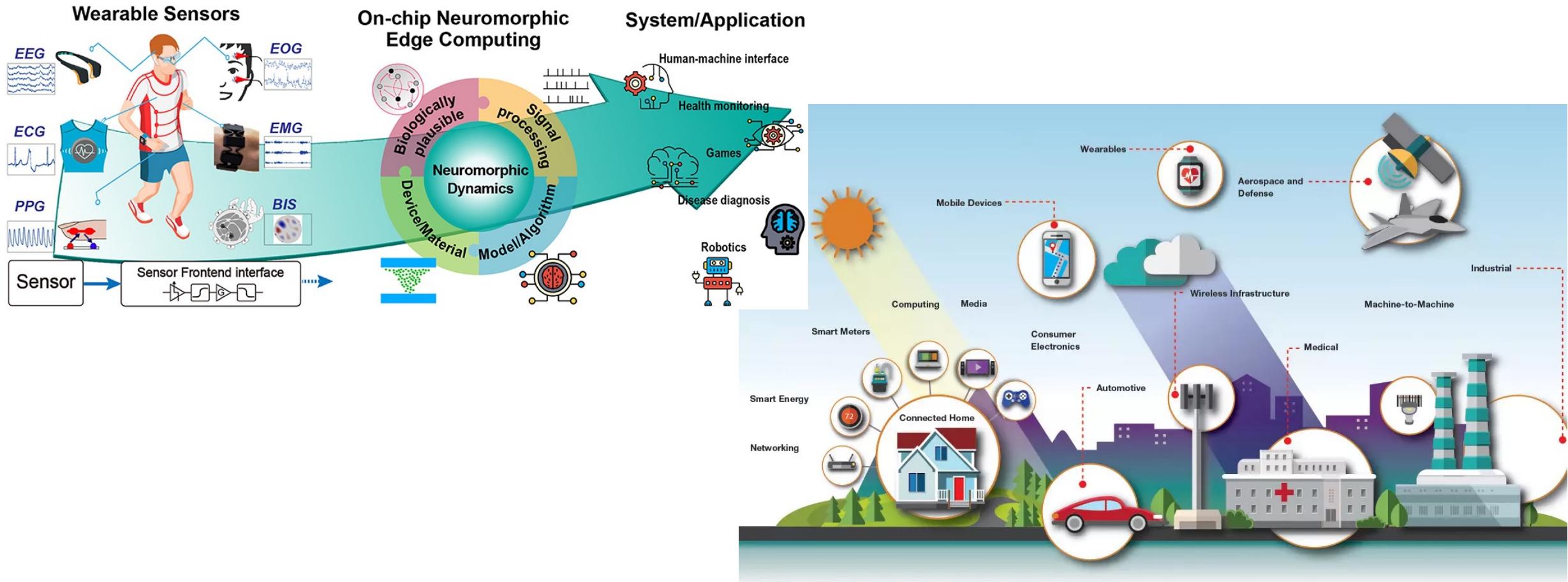


인하대학교  
INHA UNIVERSITY

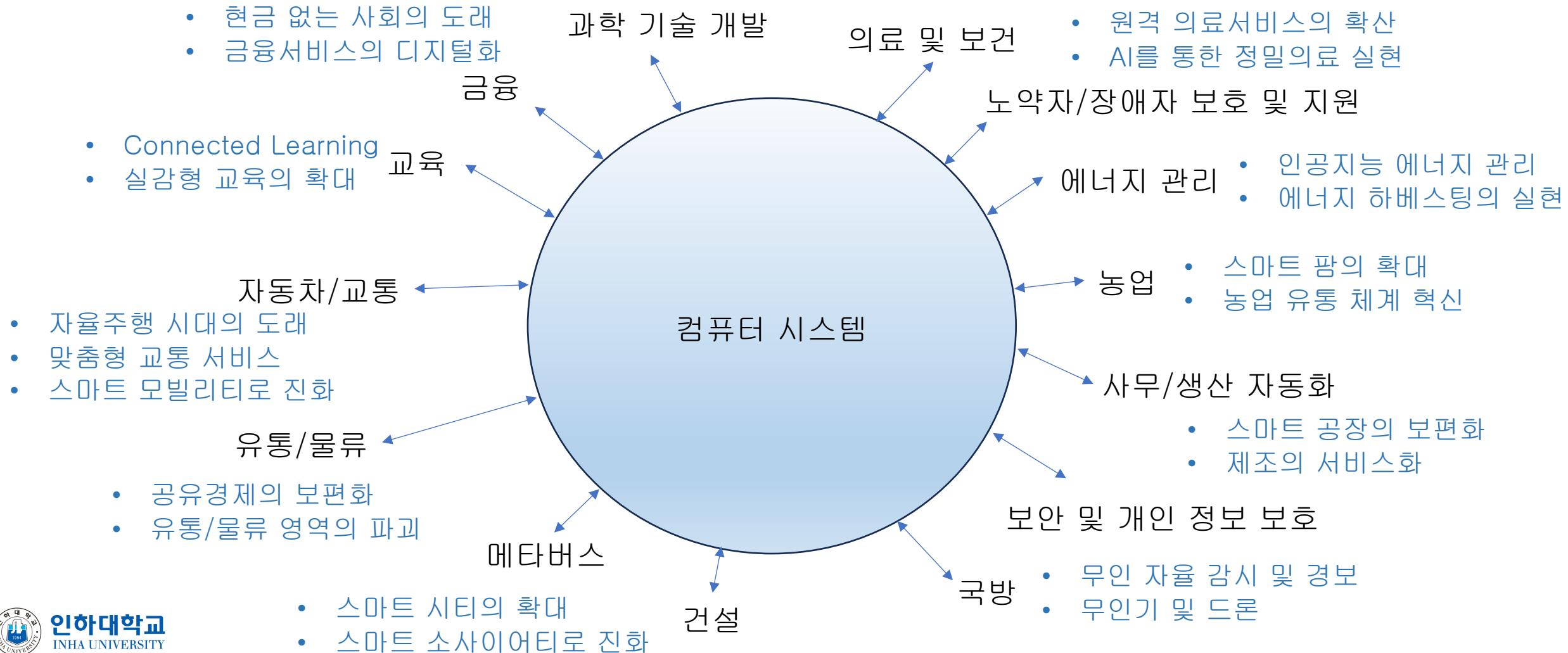
# 미래의 컴퓨터 시스템



# 미래의 컴퓨터 시스템

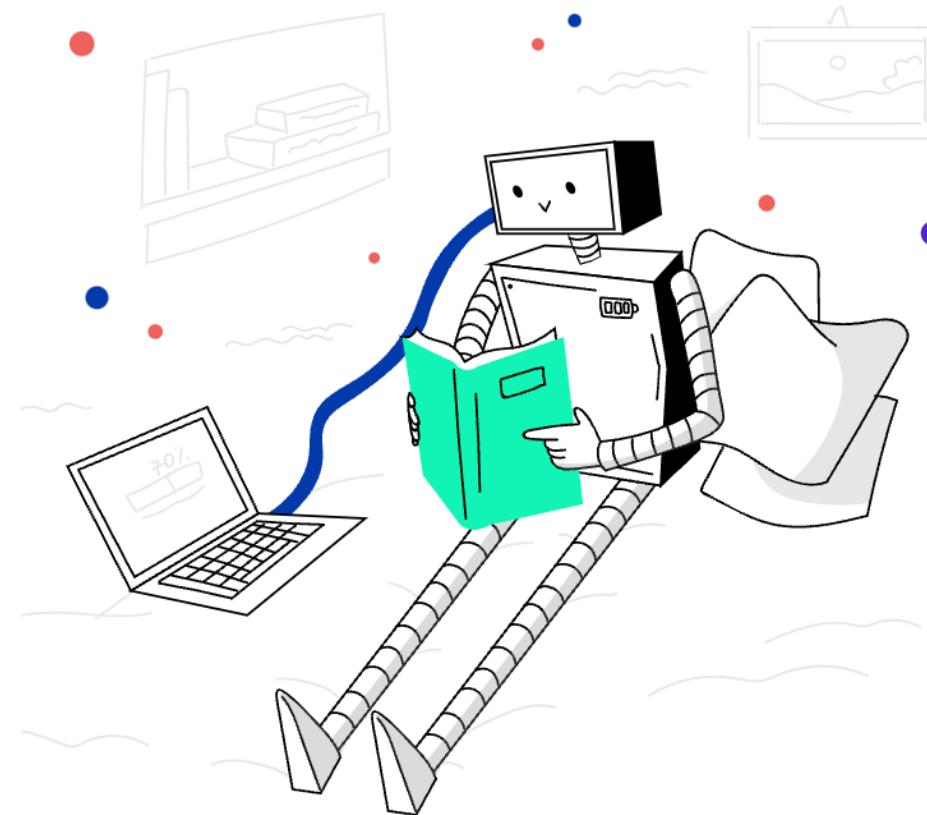


# 미래의 컴퓨터 시스템



# 컴퓨터 공학과

- Department of Computer Science and Engineering (CSE)
- 컴퓨터과학 및 컴퓨터공학을 공부하는 학과
- 컴퓨터(과/공)학과, 전산(과)학과 등으로 불림
- 근본적인 관심사는 자동화할 수 있는 것과 없는 것을 결정하고 효율적으로 자동화 하기 위한 도구를 개발하는 것



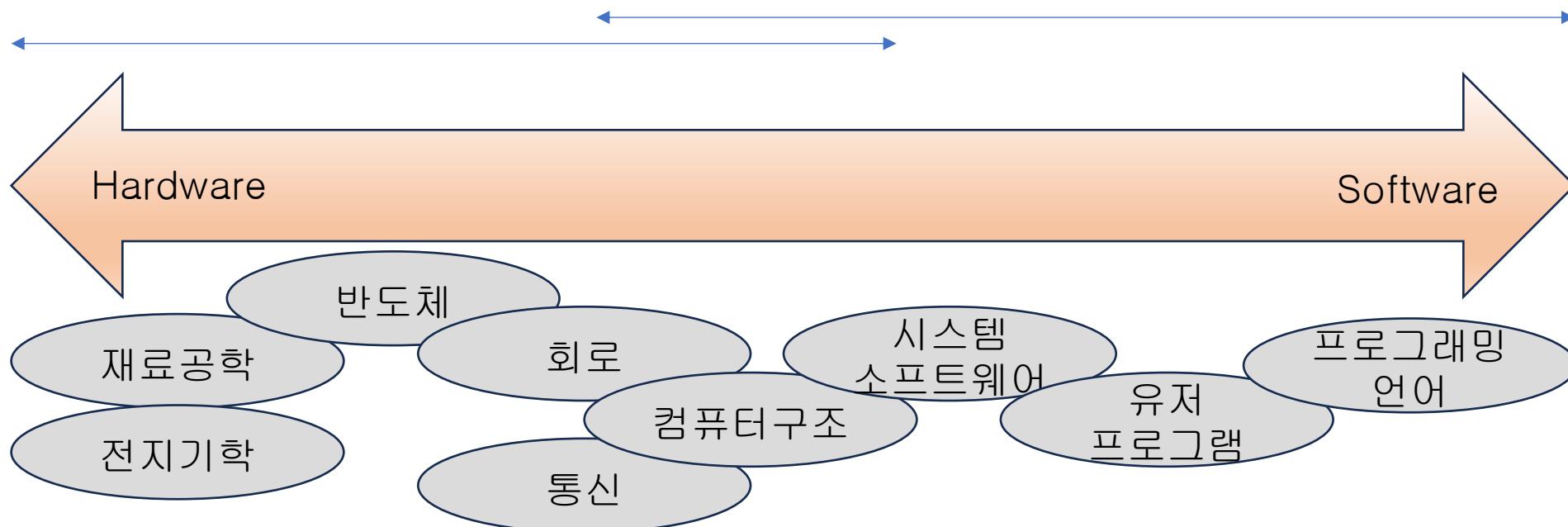
# 컴퓨터 공학과

- 이론적 분야와 실용적 분야로 나뉨
  - 실용적 분야: 하드웨어와 소프트웨어의 시스템 구현에 대한 문제들
    - 컴퓨터 그래픽스이나 계산 기하학은 보다 구체적인 응용을 강조
    - 시스템 소프트웨어 (운영체제, 분산 시스템) 등도 이론을 활용 한 구체적인 응용 및 구현을 강조
  - 이론적 연구: 알고리즘, 계산 및 정보에 대한 문제들
    - 계산 이론 – 추상적인 계산 모델과 그것들을 사용하여 해결할 수 있는 일반적인 종류의 문제에 관해 다룬다
    - 프로그래밍 언어 이론 – 계산 프로세스에 대하여 논리적으로 설명, 컴퓨터 프로그래밍은 복잡한 시스템을 만들기 위해 그것들을 사용하는 것을 포함한다. 컴퓨터 구조는 컴퓨터 구성요소와 컴퓨터 작동원리를 설명
  - 융합 분야: 인공지능은 인간과 동물에게서 발견되는 문제 해결, 의사결정, 환경 적응, 계획, 학습과 같은 목표 지향적인 과정을 종합하는 것을 목표로

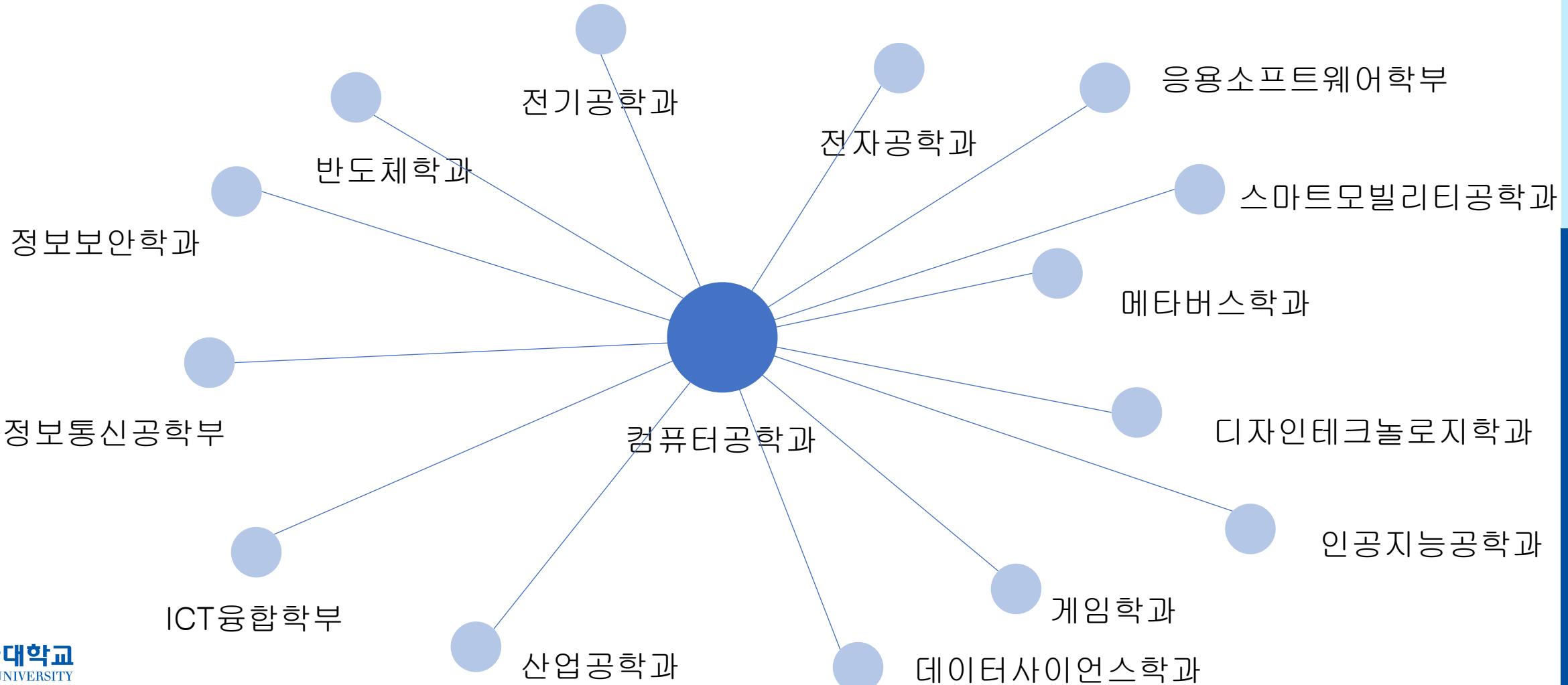
# 컴퓨터 공학과

Electrical and Electronic Engineering  
(전기전자공학과)

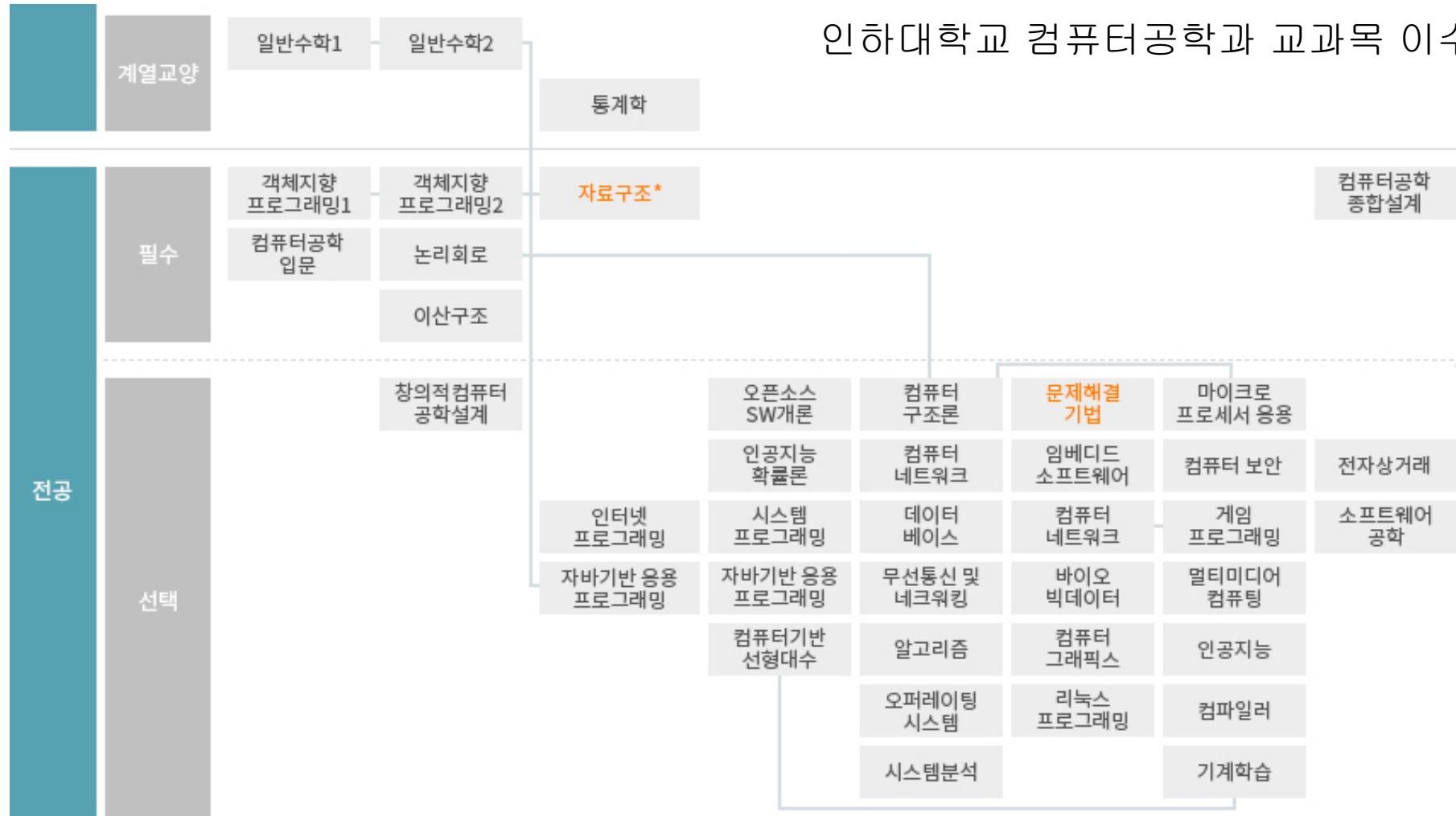
Computer Science and Engineering  
(전산학과/컴퓨터공학과)



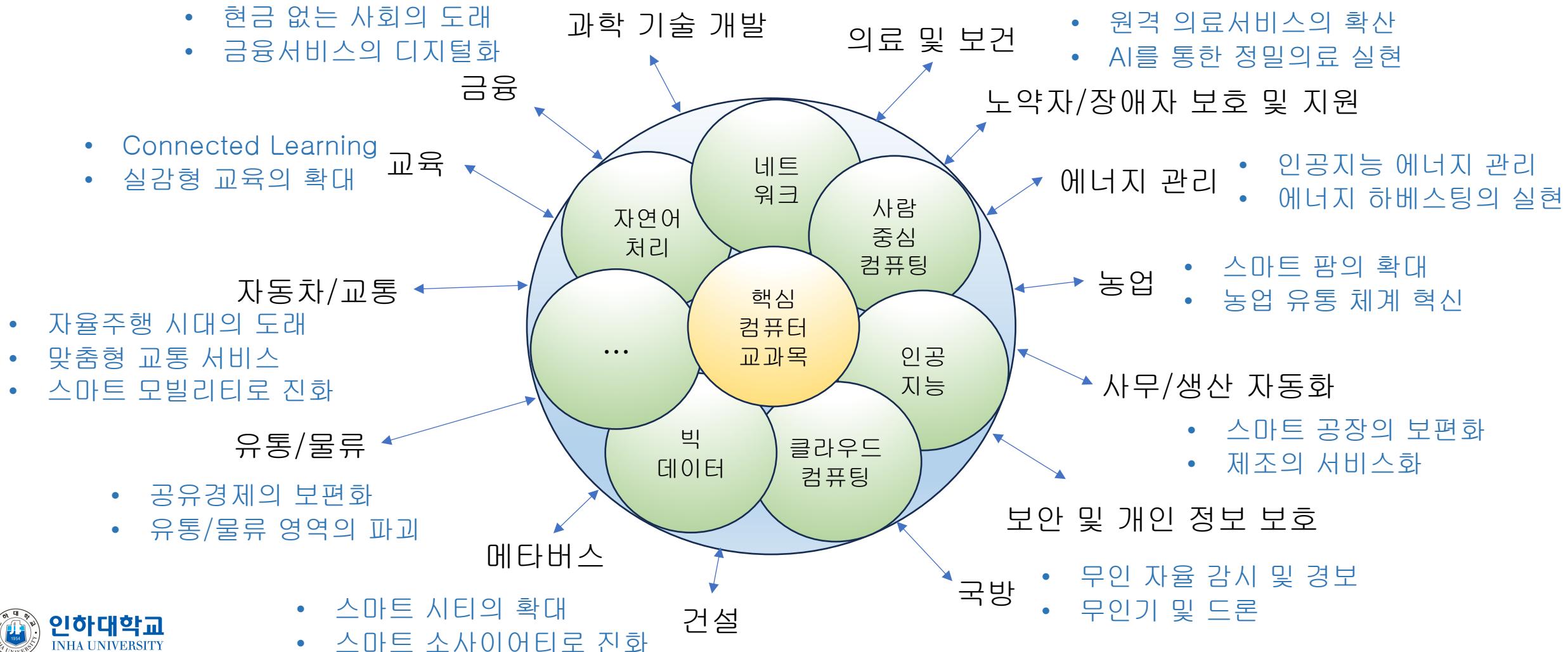
# 컴퓨터 공학과



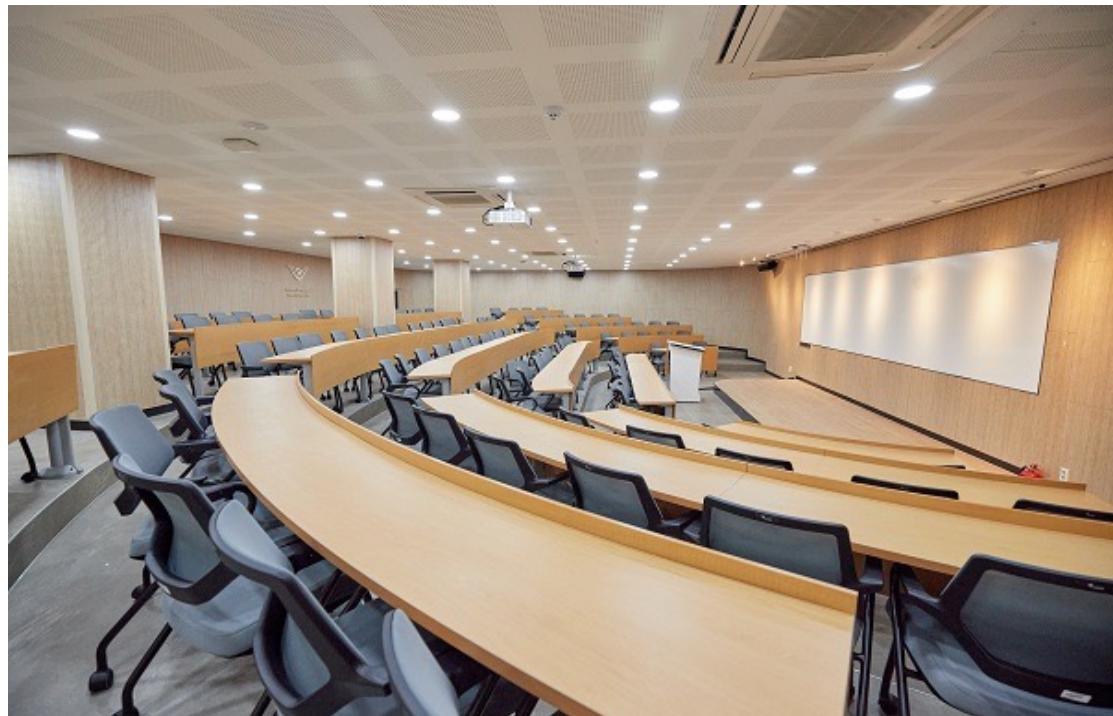
# 컴퓨터 공학과



# 컴퓨터 공학과



# 대학 생활



	월	화	수	목	금	
1교시		공업수학 1				오전 9시
2교시						
3교시						
4교시	컴퓨터기반선형대수	자료구조				오전 10시
5교시						
6교시						
7교시		컴퓨터기반선형대수		자료구조		오후 12시
8교시						
9교시						
10교시	일반수학2		일반수학2	공업수학 1		오후 1시
11교시						
12교시						
13교시		초급일본어				오후 3시
14교시						
15교시						
16교시	논리회로					
17교시						
18교시						
19교시				논리회로		오후 5시
20교시						
21교시						오후 6시
						오후 7시



인하대학교  
INHA UNIVERSITY

# 대학 생활



인하대학교  
INHA UNIVERSITY

# 대학 생활

올겨울,  
현장.실.습  
바로 도전!

2023학년도  
현장실습 설명회

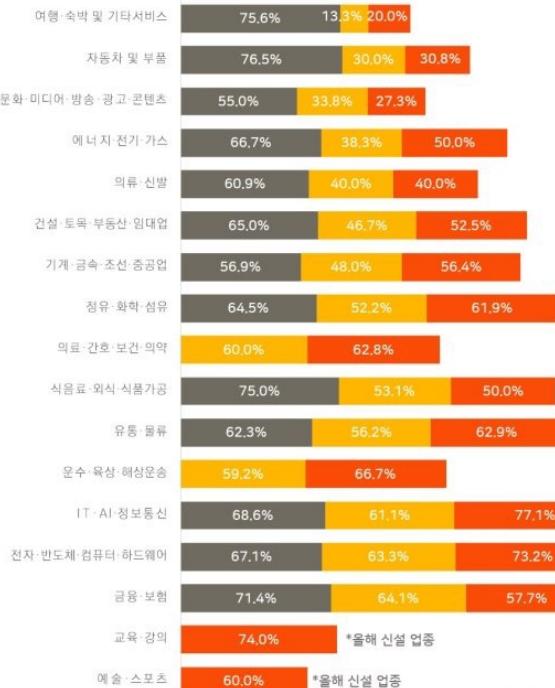
인하대학교  
INHA UNIVERSITY

# 졸업 후 진로

## 2021년 하반기 업종별 대졸신입 채용계획

조사대상: 설문 참여 기업 814곳 대상 | 조사기간: 7월 13일~8월 3일 (약 20일간)

● 2019 하반기 ○ 2020 하반기 ▲ 2021 하반기



Incruit

## 개발자 확보 전쟁, 기업 64% IT 인재 채용 어려움 겪어!

기업 383개사 설문조사 [자료제공: 사람인]

어려울 겪는다 **64.2%** 어려움 없다 **35.8%**

\* IT 인력 채용에 어려움을 겪는 이유(복수응답)

연봉 인상 등 개발자 확보경쟁이 치열해서 **50%**

능력을 갖춘 인재를 뽑기 어려워서 **47.2%**

지원자 수가 적어서 **45.5%**

회사의 인지도가 낮아서 **37%**

기업 간 치우적 수준 양극화 심해서 **17.5%**

saramin



HYUNDAI  
MOBIS

HYUNDAI  
AutoEver

WNDRVR



## 중기부와 한국 스타트업 육성 중인 글로벌 기업들

글로벌기업	프로그램명	대상분야	모집규모
Google	구글	창구	앱·게임 개발 80개사
NVIDIA	엔비디아	N&UP	AI·빅데이터 30개사
amazon web services	아마존웹서비스	정글	E커머스, 핀테크, 스마트물류 30개사
Microsoft	MS	마중	바이오헬스, 자율주행, AI 30개사
DASSAULT SYSTEMES	다쏘시스템	다온다	제조업 30개사

\*자료: 중소벤처기업부  
그래픽: 이승현 디자인기자



인하대학교  
INHA UNIVERSITY

# 졸업 후 진로

- 개발자
  - 웹 서비스, 하드웨어 펌웨어, IT 플랫폼, 임베디드 시스템, IoT, 물류 등
- 프로젝트/프로덕트 매니저
- 컴퓨터 과학자
  - 대학원 진학
  - 대학교, 국가 연구소, 기업 연구소 등
- 인공지능 전문가
  - 비전, 자율주행, 자연어 및 음성 처리 등
- 데이터 과학자
- 게임 개발자 및 기획자
- UI/UX 디자이너 및 엔지니어
- 커스터머 엔지니어
- 서버 네트워크 및 정보통신 전문가
- 정보 보안 전문가
- 테크 라이터, 기자
- 기술 고시, 공무원
- IP 전문가 및 변리사
- 정보 시스템 감사 및 인증 업무
- 컨설턴트
- 금융 및 금융 투자 회사
  - 금융결제원, 은행 IT팀, 헤지 펀드 등
- 기타 회사 IT 직군 들



# 질문?

AI