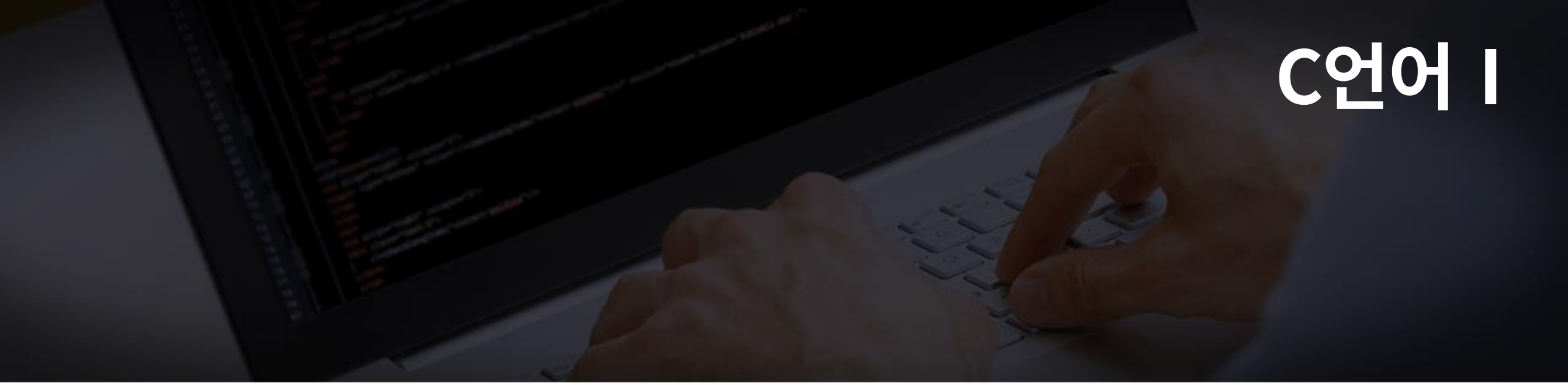
C언어 I 1주 1차시

차시명	C언어의 개요와 프로그래밍의 개념



C언어 I

Lesson 01

C언어 정의와 역사

1) C언어 정의

> 1969년과 1973년에 걸쳐서 AT&T의
> 벨 연구소에서 데니스 리치(Dennis Ritchie)에 의해
> 만들어진 고급 언어

"C"라고 하는 이름이 만들어진 이유는
직전에 만들어진 언어의 이름이 "B"였기 때문임

2) C언어 역사

C언어는 UNIX라는 운영체제의 탄생과도 밀접한 관계가 있음

C언어는 UNIX 운영 체제를 개발하기 위하여 만들어졌음

1973년에 어셈블리 언어로 작성된 PDP-II의 UNIX 커널을
C언어로 다시 작성함

2) C언어 역사

C언어의 개발동기

- 켄 톰슨(Ken Thomson)과 데니스 리치가 UNIX와 C언어를 개발하게 된 동기로 알려진 바로는 'Space Travel'이라고 하는 게임을 하기 위해서였음
- 'Space Travel'은 메인프레임용 너무 느렸기 때문에 PDP-II이라는 사무실에 있었던 미니 컴퓨터로 이식하기로 하였음
- 하지만 PDP-11은 운영체제가 없었고 따라서 운영 체제를 작성하기로 하였는데 모든 코드를 어셈블리 언어로 작성하기에는 너무 어렵고 지루하였다고 함

2) C언어 역사

C언어의 개발동기

- 두 사람은 운영 체제를 고수준의 언어로 작성하여 운영체제가 한 번 작성되면 다른 컴퓨터에 쉽게 이식하는 것이 필요하다고 생각하게 되었고 언어 “B”에 관심을 가짐

- 그러나 언어 “B”는 당시 최신 컴퓨터였던 PDP-11의 성능을 충분히 활용하기에는 모자란 점이 있다고 판단하고 새로운 언어 “C”를 개발하였음

3) C언어의 다양성

K&RC

- 1978년에 리치(Ritchie)와 브라이언 커닝햄(Brian Kernig han)은 “C Programming Language” 도서를 발간하였는데, 이 책에서 사용한 버전임

ANSIC

- 1983년에 ANSI(American National Standards Institute)는 X3J11이라는 위원회를 만들고 C언어의 표준을 만들게 됨

3) C언어의 다양성

C99

- 1999년에 ISO는 C언어에 대한 새로운 표준을 다시 공표함
  ➡ 이것은 C99로 불림
- C99에서는 C++에서 널리 사용되고 있던 여러 가지 특징들을 추가하였음

C11

- ISO에 의하여 2011년 12월에 발표된 C언어 표준임

4) C언어의 특징

1 C언어는 간결한 언어임

- C언어에는 꼭 필요한 기능만이 들어 있고 모든 표기법이 아주 간결하게 되어 있음
- 간결성은 C언어의 핵심적인 특징임

2 C언어는 효율적인 언어임

- C언어는 효율적인 언어임
- 효율적이라는 의미는 C로 작성된 프로그램이 크기가 작으며 실행 속도가 빠르고 메모리를 효과적으로 사용한다는 것을 의미함
- C언어는 거의 어셈블리 언어 수준의 효율성을 자랑함

  ➡ 이점은 상업용 프로그램을 작성할 때 큰 장점이 됨

4) C언어의 특징

3 C언어는 저수준/고수준의 프로그래밍이 모두 가능함

- C언어는 운영 체제를 만들었던 언어이니 만큼,
  어셈블리 언어의 구체적인 하드웨어 제어가 가능함
- 실제로 스마트폰 TV 세탁기 등의 여러 가지 전자 기기 안에 들어가는
  임베디드(내장) 프로그램은 대부분 C언어로 개발됨

예 안드로이드 폰의 운영체제는 리눅스로를 기반으로 한 C언어로
작성되어 있음

4) C언어의 특징

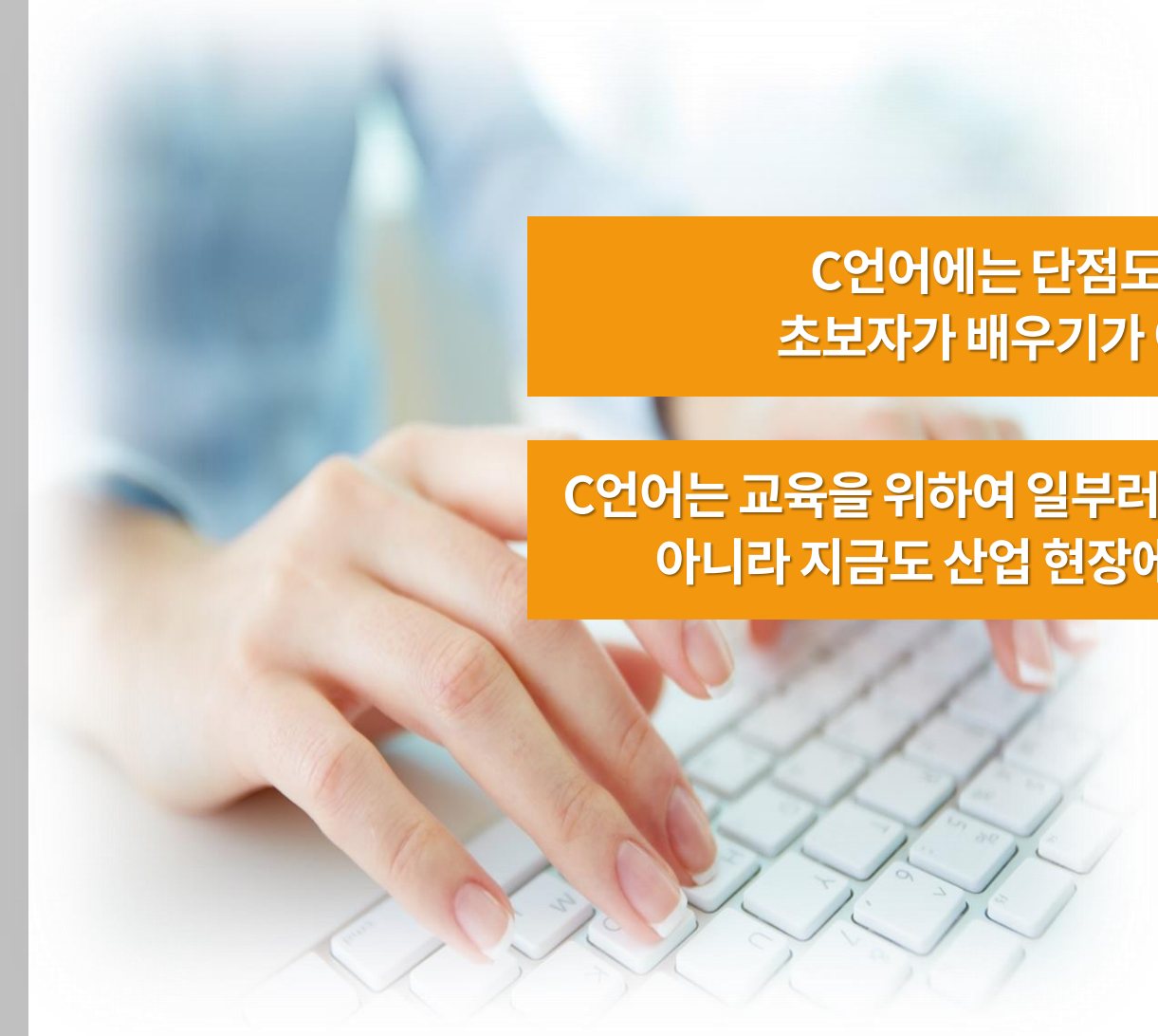
4	C언어는 이식성이 뛰어남

이식성 (Portability)	한번 작성된 프로그램을 다른 CPU를 가지는 하드웨어로 쉽게 이식할 수 있다는 의미

- 많은 종류의 CPU에 대하여 C 컴파일러가 개발되어 있기 때문에 C 프로그램은 상대적으로 이식성이 좋음
- PC에서 개발된 프로그램도 컴파일만 다시 하면 슈퍼 컴퓨터에서도 수행시킬 수 있음

4) C언어의 특징

C언어의 단점



C언어에는 단점도 존재하는데
초보자가 배우기가 어렵다는 것임

C언어는 교육을 위하여 일부러 쉽게 만들어 놓은 언어가
아니라 지금도 산업 현장에서 사용되는 언어임

5) 알고리즘

사례

전화번호부에서 김영이라는 사람의 전화번호를 찾는 문제를 알고리즘을 적용할 경우

1	전화번호부의 첫 페이지부터 시작하여 한 장씩 넘기면서 김영을 찾는 것

- 이 방법은 엄청난 시간이 걸리는 방법이고 보통 이런 식으로 찾는 사람은 거의 없음

5) 알고리즘

사례

전화번호부에서 김영이라는 사람의 전화번호를 찾는 문제를 알고리즘을 적용할 경우

2	전화번호부의 이름들이 정렬되어 있음을 이용하는 방법

- 전화번호부의 중간 정도를 펼쳐서 거기에 있는 이름들과 김영을 비교하여 앞부분으로 가던지 뒷부분으로 감
- 계속하여 다시 찾아야 할 범위의 중간 부분에 있는 이름과 김영을 비교함
- 이러한 과정을 김영이란 이름을 찾을 때까지 되풀이함

5) 알고리즘

사례

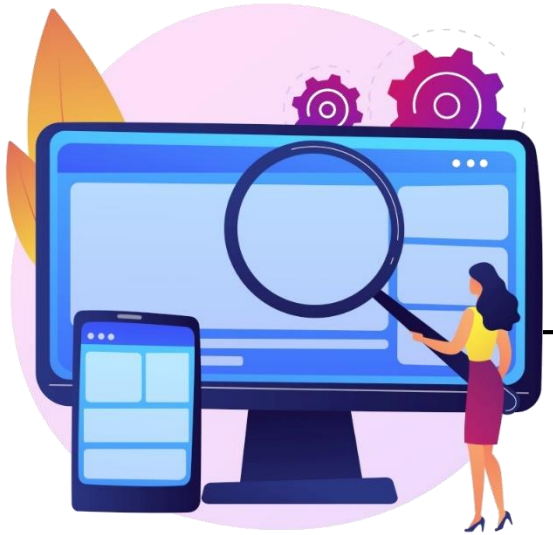
전화번호부에서 김영이라는 사람의 전화번호를 찾는 문제를 알고리즘을 적용할 경우

1	전화번호부의 첫 페이지부터 시작하여 한 장씩 넘기면서 김영을 찾는 것
2	전화번호부의 이름들이 정렬되어 있음을 이용하는 방법

- 이러한 방법은 프로그래밍 언어와는 무관함
- C언어를 사용하든, Java 언어를 사용하든, 파이썬 언어를 사용하든 문제를 풀어나가는 방법은 동일함

5) 알고리즘

정의



알고리즘

주어진 문제를 풀기 위하여 컴퓨터가
수행하여야 할 단계적인 절차를 기술한 것

알고리즘을 프로그래밍 언어로 구현하면
프로그램이 됨

5) 알고리즘

알고리즘의 비유

5) 알고리즘

알고리즘의 비유

빵을 만드는 알고리즘

1	빈 그릇을 준비함
2	이스트를 밀가루, 우유에 넣고 저음
3	버터, 설탕, 계란을 추가로 넣고 섞음
4	따뜻한 곳에 놓아두어 발효시킴
5	170~180도의 오븐에서 구움

5) 알고리즘

알고리즘의 비유

빵을 만드는 알고리즘

- 빵을 만들 때도 순서가 잘못되면 빵이 만들어지지 않음
- 빵을 만드는 방법은 영어, 독일어 프랑스어로도 정확하게 표현할 수 있듯이 알고리즘은 어떤 프로그래밍 언어로도 동일하게 표현할 수 있음
- 같은 빵을 만드는 방법도 여러 가지가 존재할 수 있듯이
  하나의 문제에 대하여 알고리즘은 여러 가지가 존재할 수 있음

  ➡ 이 경우 프로그래머는 가장 효율적인 알고리즘을 선택하여 구현하여야 할 것임

5) 알고리즘

알고리즘의 예

1부터 10까지의 합을 구하는 문제

Case 1	Case 2	Case 3

- 1부터 10까지의 숫자를 직접 하나씩 더함

$$1+2+3+\ldots+10=55$$

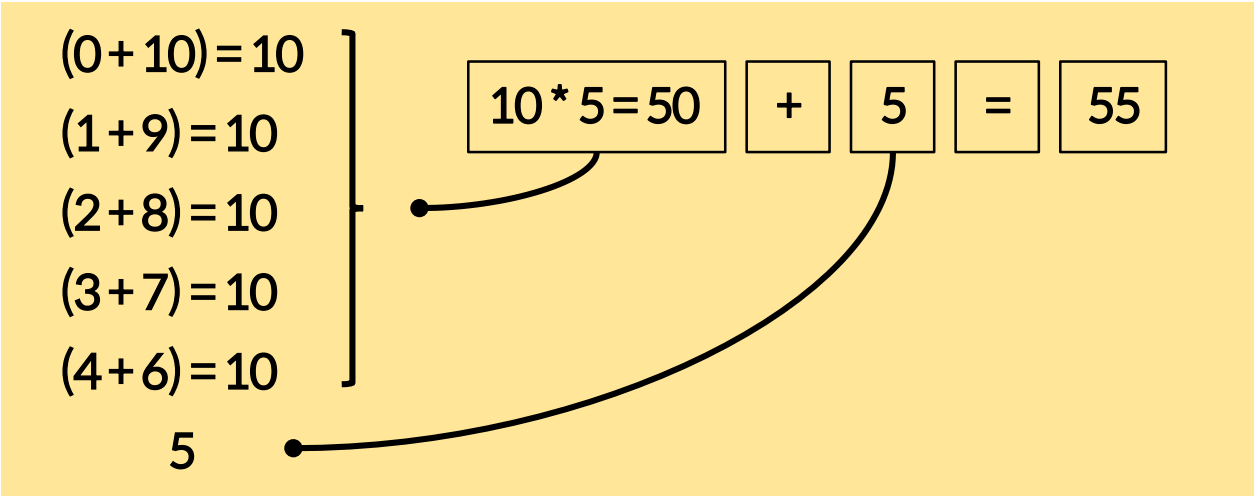
5) 알고리즘

알고리즘의 예

1부터 10까지의 합을 구하는 문제

Case 1 | **Case 2** | Case 3

- 두수의 합이 10이 되도록 숫자들을 그룹핑하여 그룹의 개수에 10을 곱하고 남은 숫자 5를 더함



5) 알고리즘

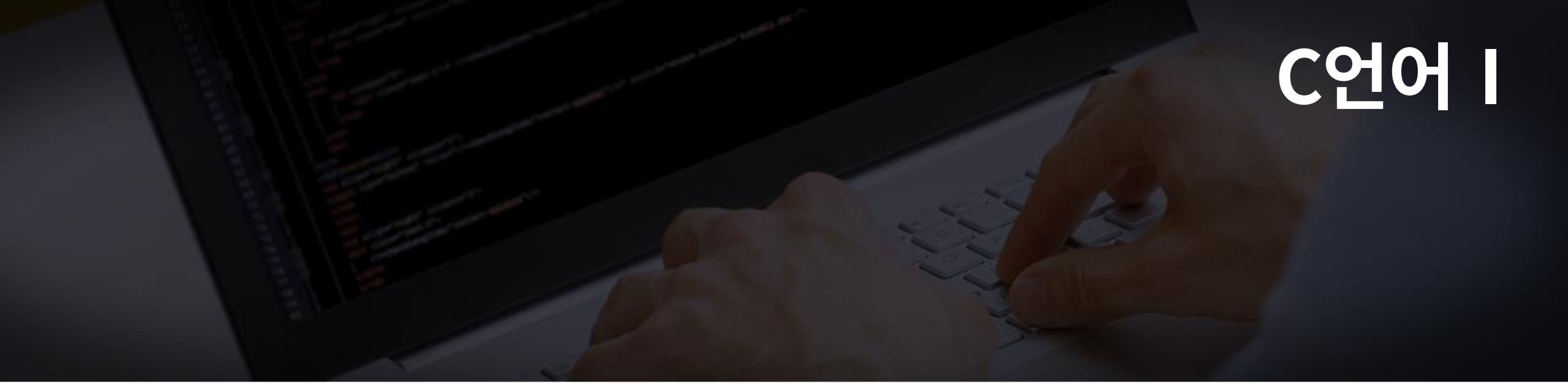
알고리즘의 예

1부터 10까지의 합을 구하는 문제

Case 1	Case 2	**Case 3**

- 공식을 이용하여 계산할 수도 있음

$$10(1+10)/2=55$$



C언어 I

Lesson 02

프로그래밍의 개념

김영평생교육원

1) 프로그래밍이란

프로그램의 중요성

컴퓨터를 사용하여 처리할 수 있는 작업

- 문서 작성
- 회계 장부 정리
- 사진 편집

➡ 이러한 작업들은 하드웨어만 있다고 할 수 있는 일이 아님

1) 프로그래밍이란

프로그램의 중요성

이러한 작업들이 가능하기 위해서는 응용 프로그램을 설치해야 함

문서 작성	'한글'과 같은 워드프로세서 프로그램
회계 장부 정리	'엑셀'과 같은 스프레드시트 프로그램
사진 편집	'포토샵'과 같은 이미지 편집 프로그램

1) 프로그래밍이란

프로그램의 중요성

이러한 작업들이 가능하기 위해서는 응용 프로그램을 설치해야 함

‘윈도우즈’와 같은 운영체제 설치

↓

추가로 여러 가지 응용 프로그램 설치

↓

비로소 우리가 유용하게 사용할 수 있는 컴퓨터가 되는 것

프로그램이 없다면 컴퓨터는
약간의 열과 소음을 발생하는 쓸모없는 기계에 불과함

1) 프로그래밍이란

프로그램의 중요성

그렇다면 왜 컴퓨터에서는 가전제품처럼
프로그램 설치 없이 바로 동작되도록 하지 않고
불편하게 사용자가 프로그램을 설치하게 하였을까

▶ 컴퓨터가 가전제품과 다른 점은 컴퓨터는 범용적인 기계라는 점임

가전제품	컴퓨터
미리 정해진 한 가지 작업 밖에 못함	프로그램만 바꿔주면 매우 다양한 작업을 할 수 있음

1) 프로그래밍이란

프로그램의 중요성

그렇다면 왜 컴퓨터에서는 가전제품처럼
프로그램 설치 없이 바로 동작되도록 하지 않고
불편하게 사용자가 프로그램을 설치하게 하였을까

- 스마트폰이 기존의 휴대폰에 비해서 인기가 있는 이유도 자신에게 필요한 각종 앱들을 자유롭게 설치할 수 있기 때문임
  - 스마트폰 때문에 MP3 플레이어, 카메라, 전자사전, 내비게이션, 휴대용 게임기 등의 매출이 큰 타격을 받았음

1) 프로그래밍이란

프로그램의 중요성

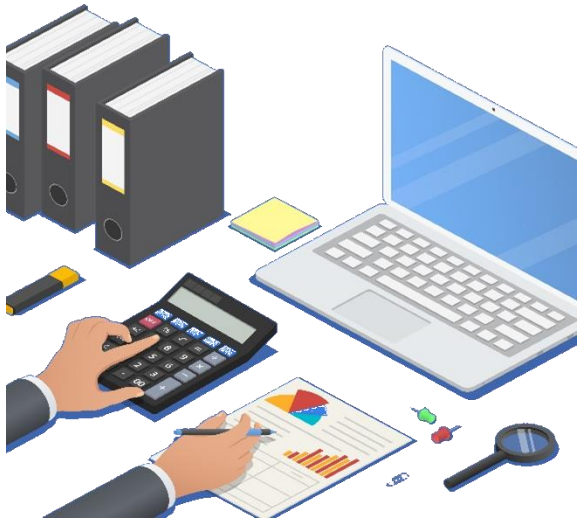
그렇다면 왜 컴퓨터에서는 가전제품처럼
프로그램 설치 없이 바로 동작되도록 하지 않고
불편하게 사용자가 프로그램을 설치하게 하였을까

- 동일한 하드웨어에서 프로그램을 바꾸어가면서 다양한 작업을 할 수 있는 것은 컴퓨터의 가장 강력한 장점임
  - 컴퓨터에 프로그램이 없으면 우리는 아무런 일도 할 수 없음
  - 컴퓨터는 근본적으로 미리 작성된 프로그램을 단순히 수행하는 기계인 것임

1) 프로그래밍이란

컴퓨터 Vs. 계산기

컴퓨터의 기본적인 임무는 숫자 계산을 빠르게 하는 것임



But 계산만 빨리 할 수 있다고 해서
컴퓨터라고 부를 수 있을까?

계산기도 계산을 빠르게 하지만
아무도 계산기를 컴퓨터라고 부르지 않음

1) 프로그래밍이란

컴퓨터 Vs. 계산기

요즘의 컴퓨터는 계산만 하는 기계는 아님

현대적인 의미의 컴퓨터

명령어들의 리스트에 따라 데이터를 처리하는 기계

프로그램 (Program)	특정한 작업을 수행하도록 설계된 명령어들의 리스트

1) 프로그래밍이란

컴퓨터 Vs. 계산기

- 다양한 프로그램을 수행할 수 있는 능력은 컴퓨터를 다재다능한 기계로 만들었으며 이것이 바로 계산기와 컴퓨터를 구별하는 중요한 특징이 됨

- 스마트폰 시대가 온 것도 자신이 필요한 앱을 스마트폰에 설치할 수 있는 특징 때문임

  → 앱(Application: 응용 프로그램)이 바로 프로그램임

1) 프로그래밍이란

컴퓨터 Vs. 계산기

프로그램이라는 개념을 도입하여
수행하는 기능을 쉽게 변경할 수 있음

정해진 기능만을 수행하며,
기능을 변경할 수 없음

1) 프로그래밍이란

프로그램 안에 들어 있는 것

- 프로그램은 우리가 하고자 하는 작업을 컴퓨터에게 전달하여 주는 역할을 함
- 프로그램은 특정한 작업을 위한 작업 지시서라고 보면 됨
- 작업을 지시하려면 명령어(Instruction)들을 나열해야 함
- 프로그램 안에는 명령어들이 들어 있음

1) 프로그래밍이란

프로그램 안에 들어 있는 것

로봇에게 가까운 햄버거 가게에 가서 햄버거 사오는 일을 시킨다고 가정할 경우

**이 일은 다음과 같은 지시사항들로 이루어질 수 있는데,
이 지시사항들이 바로 명령어임**

① 500미터 직진함
② 교차로에서 우회전함
③ 1,000미터 직진함
④ 왼쪽에서 햄버거 가게를 찾음
⑤ 햄버거를 주문함
⑥ 햄버거를 들고 출발한 위치로 다시 옴

1) 프로그래밍이란

프로그램 안에 들어 있는 것

로봇에게 가까운 햄버거 가게에 가서 햄버거 사오는 일을 시킨다고 가정할 경우

- 컴퓨터에서의 명령어는 로봇과 약간 다름

컴퓨터에서의 명령어는 주로 “두 수를 더하라” 또는 “메모리에서 CPU로 데이터를 이동하라”, “입력장치에서 데이터를 읽어라”와 같은 것으로 구성됨

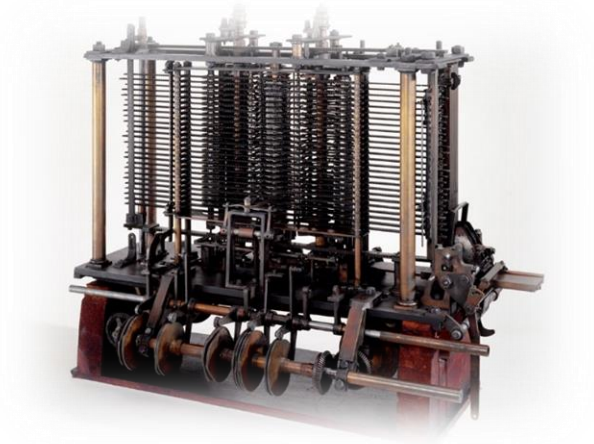
1) 프로그래밍이란

프로그램의 역사

해석 기관(Analytical Engine)

최초의 기계식 컴퓨터는 18세기에 “컴퓨터의 아버지”로 불리는
찰스 배비지가 설계한 **“해석 기관(Analytical Engine)”**이라고 함

**찰스 배비지는 복잡한 수학 연산을 기계로 풀어낼 수 있다고 주장하였고
모든 종류의 계산을 하나의 기계에서 할 수 있도록 설계된
최초의 범용 컴퓨터였음**



[출처: 위키백과]

1) 프로그래밍이란

프로그램의 역사

해석 기관(Analytical Engine)

- 해석 기관에서 프로그램을 수행하는 것이 가능했다는 것이 특징임

  But 해석 기관은 개념과 원리만 완성됐을 뿐, 실제로 만들어지지 못했음

- 배비지의 해석 기관은 수천 개의 기어, 바퀴, 축, 레버 등으로 구성하여 증기로 작동하도록으로 설계되었음

- 해석 기관에는 현대 디지털 컴퓨터의 기초적인 하드웨어와 소프트웨어픽 원리가 구현돼 있었음

1) 프로그래밍이란

프로그램의 역사

해석 기관(Analytical Engine)

- 해석 기관은 현대 컴퓨터에서도 사용하는 네 가지 핵심적인 부품(Component)이 모두 포함되어 있었음

중앙 처리 장치	계산을 담당, Mill이라고 불림
메모리	중간 단계에서 임시적으로 숫자가 저장, Store라고 불림
출력 장치	출력 숫자를 나타내는 다이얼
입력 장치	천공 카드

1) 프로그래밍이란

프로그램의 역사

에니악(ENIAC)

에니악(ENIAC)

최초의 전자식 컴퓨터는 1943년에 탄도 궤적을 계산할 목적으로 개발되었으며, 18,000개의 진공관과 6,000여 개의 스위치로 이루어져 있었음

- 10진법을 사용해서 계산을 하는 최초의 범용 전자 컴퓨터였음
- 산술 연산과 논리 연산을 수행할 수 있었고 변수 개념도 지원하였음

1) 프로그래밍이란

프로그램의 역사

에니악(ENIAC)

결정적인 문제점
▪ 에니악은 탄도 궤도 표를 계산하는 특수 목적 컴퓨터였기 때문에 설계된 목적만을 수행할 수 있었고 다른 작업을 수행시키려면 아주 복잡하였음 ▪ 에니악의 프로그램은 스위치에 의하여 기억되었고 프로그램을 변경할 때 마다 그 많은 스위치들을 처음부터 다시 연결하여야 했다고 함

1) 프로그래밍이란

프로그램의 역사

프로그램 내장(Stored Program) 구조 또는 폰 노이만 구조

에니악의 문제점을 개선하기 위한 노력의 산물

- 이 프로그램 내장 방식은 폰 노이만이 1945년에 발표된 논문에서 최초로 기술되었음
- 프로그램 내장 구조란 프로그램을 메모리에 저장하는 방식임
- 컴퓨터가 기계의 메모리에 적재된 프로그램에 의해 작동될 수 있다는 것임

1) 프로그래밍이란

프로그램의 역사

프로그램 내장(Stored Program) 구조 또는 폰 노이만 구조

에니악의 문제점을 개선하기 위한 노력의 산물

- 프로그램이 저장 장치에 내장되어 반복적으로 메모리에 적재될 수 있고, 프로그램 자체도 다른 데이터와 마찬가지로 수정될 수 있었음

  ➡ 프로그램을 데이터처럼 취급하는 개념이었음

1) 프로그래밍이란

프로그램의 역사

프로그램 내장(Stored Program) 구조 또는 폰 노이만 구조

특징
▪ 프로그램과 데이터가 모두 메인 메모리에 저장됨 ▪ 메인 메모리에 저장된 프로그램에서 2진수로 되어 있는 명령어들을 순차적으로 가져와서 실행함

1) 프로그래밍이란

프로그램의 역사

에드박(EDVAC)

에드박(EDVAC)

최초의 실용적인 프로그램 내장 방식의 컴퓨터로 1948년에 제작되었음

- 에드박은 최초로 메모리를 가지고 있었고 이 메모리 안에 프로그램을 내장하여 수행하였음
- 에드박은 0과 1의 이진 숫자들로 구성된 기계어를 사용하였음

1) 프로그래밍이란

프로그래머(Programmer)

프로그램을 전문적으로 작성하는 사람을
프로그래머(Programmer)라고 함

1) 프로그래밍이란

프로그래머(Programmer)

프로그램을 최초로 만든
에이다 러브레이스(Ada Lovelace)

역사상 최초의 프로그래머

낭만파 시인 바이런의 친딸로서 1833년에
배비지가 만들던 “차분 기관”과 “해석 기관”에
매료되어 천공 카드로 명령을 받아서
해석 기관에서 다양한 계산을 수행하는
프로그래밍의 개념을 개발하였음

1) 프로그래밍이란

프로그래머(Programmer)

- 배비지는 에이다를 숫자의 마술사(Enchantress of Number)라고 불렀다고 함

- 에이다는 현대적인 컴퓨터가 등장하기 100년 전에 이미 서브루틴(Subroutine), 루프(Loop), 점프(Jump) 등의 핵심적인 컴퓨터 프로그래밍 기본 원리를 고안하였음

- 미국 국방성에서는 에이다 러브레이스를 기념하기 위하여 자신들의 언어를 에이다(ADA)라고 이름지었음

2) 프로그래밍 언어

컴퓨터가 이해하는 언어

프로그램 = **컴퓨터에게 작업을 지시하는 문서**

그렇다면 어떤 언어를 사용해야만
컴퓨터가 작업 지시를 이해할 수 있을까

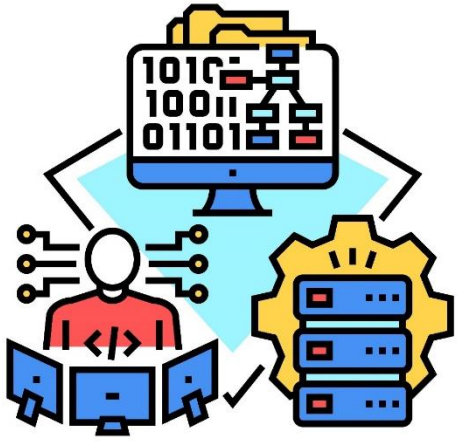
- 컴퓨터는 프로그래머가 시키는 대로만 하는 단순한 기계라고 하였으니 사람의 언어를 이해할 것 같지는 않음
- 한국어나 영어 등으로 작업을 기술한다면 컴퓨터는 전혀 이해할 수 없을 것임

2) 프로그래밍 언어

컴퓨터가 이해하는 언어

> "컴퓨터가 바로 알아듣는 언어는 한 가지"

0과 1로 구성되어 있는 "00110111000101010..."과 같은 2진수



- 컴퓨터는 2진수의 개념 위에 만들어진 기계
- 컴퓨터는 모든 것을 0과 1로 표현하고 0과 1에 의하여 내부 스위치 회로들이 ON/OFF(커짐/꺼짐) 상태로 변경되면서 작업이 진행됨
  - ➡ 0이면 회로를 끄고 1이면 회로를 켬

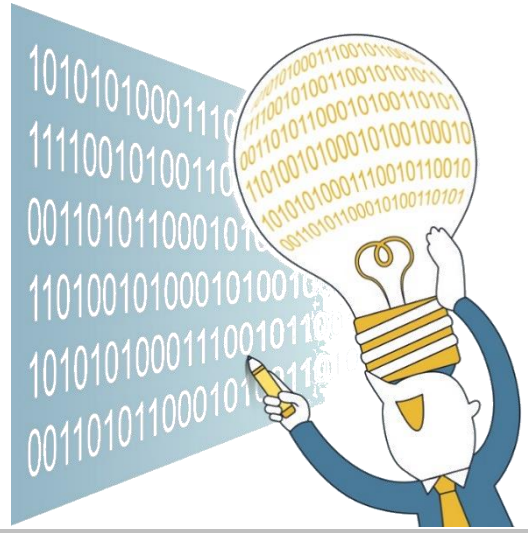
2) 프로그래밍 언어

컴퓨터가 이해하는 언어

기계어(Machine Language)

2진수 형태의 언어를 의미하며, 컴퓨터가 가장 좋아하는 언어임

실제로 초기의 컴퓨터에서는 이러한 기계어를 사용하여
프로그램을 작성하였음



2) 프로그래밍 언어

컴퓨터가 이해하는 언어

기계어는 인간에게 상당히 불편한 언어였기 때문에
좀 더 편리한 언어가 필요했고, 사람들은 점차적으로
인간의 언어에 더욱 근접한 프로그래밍 언어를 만들었음

- 프로그래밍 언어들은 기계어와 인간이 사용하는 자연어 중간쯤에 위치함

- 인간이 프로그래밍 언어를 배워서 프로그램을 작성하면 컴파일러라고 하는 통역을 담당하는 소프트웨어가 프로그램을 기계어로 바꾸어 줌
  - 영어를 말하는 사람과 한국어를 말하는 사람이 중간에 통역을 두고 이야기하는 것과 비슷함

2) 프로그래밍 언어

컴퓨터가 이해하는 언어

인간은 기계어를 학습하기에는 너무 힘들고
컴퓨터가 인간의 언어를 이해한다는 것은
아주 먼 미래의 이야기임

중간에 **통역의 역할을 하는 프로그래밍 언어**를
두고 작업을 지시하는 것임

C언어는 이러한 프로그래밍 언어의 일종임

2) 프로그래밍 언어

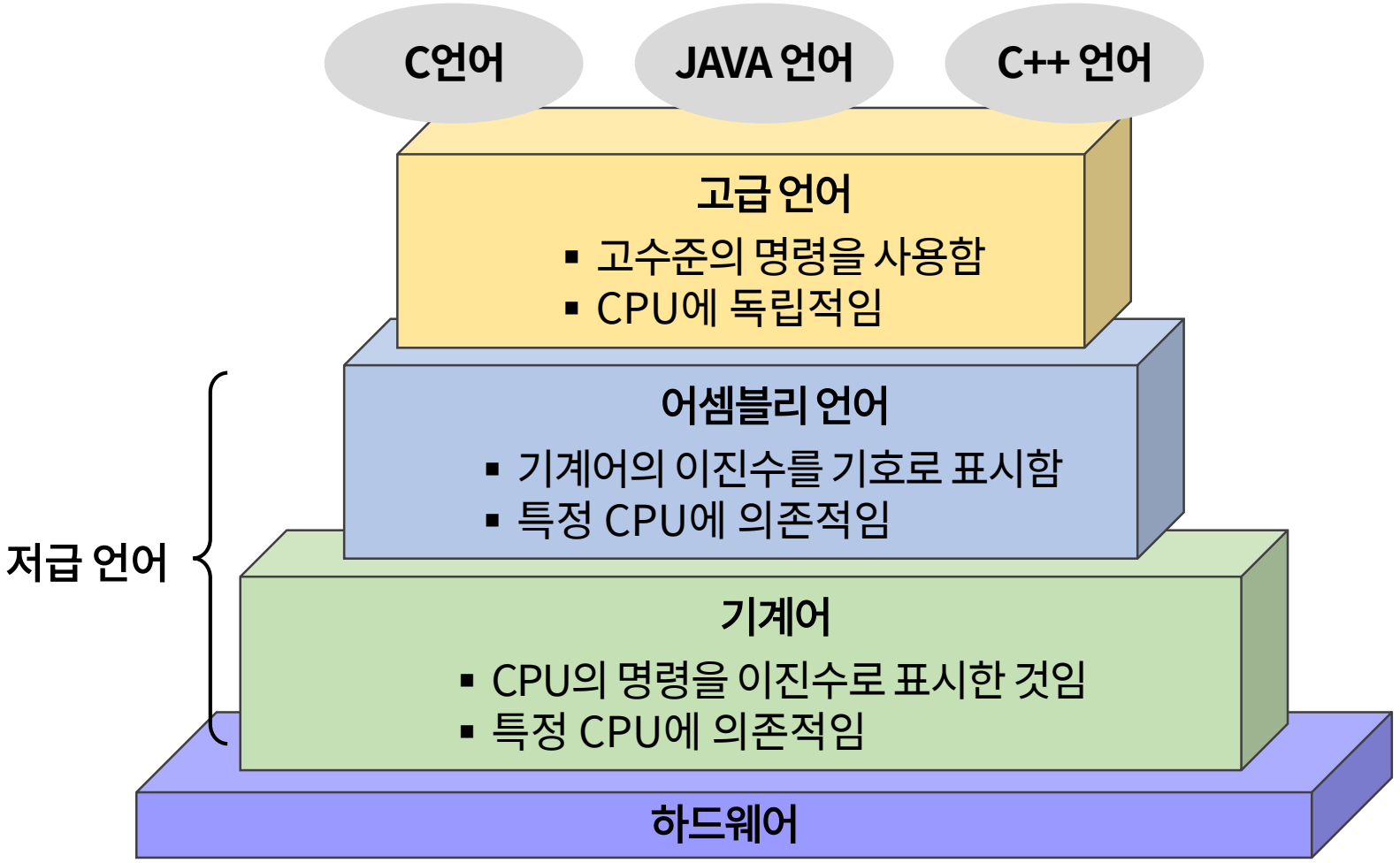
프로그래밍 언어의 분류

프로그래머들이 선택할 수 있는 프로그래밍 언어는 굉장히 많은데,
이 언어들은 대개 다음의 **세 가지로 분류**할 수 있음

- 기계어 (Machine Language)
- 어셈블리 언어 (Assembly Language)
- 고급 언어 (High-level Language)

2) 프로그래밍 언어

프로그래밍 언어의 분류



2) 프로그래밍 언어

기계어

기계어는 컴퓨터가 바로 이해할 수 있는 단 하나의 언어임

- 어떠한 프로그래밍 언어라도 전처리와 컴파일 단계를 거치면 결국은 기계어가 됨
- 기계어는 특정 컴퓨터의 명령어(Instruction)를 2진수로 표시한 것이며 컴퓨터 하드웨어를 설계할 때 결정됨
- 기계어는 하드웨어에 따라 달라지기 때문에 철저히 하드웨어에 종속됨
  - 인텔의 CPU를 사용하는 프로그램을 ARM의 CPU를 사용하는 컴퓨터에서 바로 실행시키지 못하는 이유임
- 기계어는 인간이 사용하기에는 너무 불편하고 지루함

2) 프로그래밍 언어

기계어

중간고사 성적과 기말고사 성적을 더하여
성적 합계를 구하는 연산을 가상적으로 기계어로 표시한 예

00001111	10111111	01000101	11111000
00001111	10111111	01001101	11111000
00000011	10100001		
01100110	10001001	01000101	11111010

2) 프로그래밍 언어

어셈블리 언어

기계어가 인간이 사용하기에는 **너무 힘들고 오류가 발생**하기 쉬웠으므로
프로그래머들은 **어셈블리 언어를 개발**하게 되었음

- 어셈블리 언어를 사용하면 프로그래머들은 CPU의 명령어들을 기호(Symbolic Name)로 표기할 수 있었음
- 어셈블리 언어는 프로그래머가 기계어보다는 더 높은 수준에서 프로그램을 작성하는 것을 가능하게 하였음
- 어셈블러(Assembler)라는 프로그램이 기호를 2진수로 변환함

2) 프로그래밍 언어

어셈블리 언어

기계어가 인간이 사용하기에는 **너무 힘들고 오류가 발생**하기 쉬웠으므로
프로그래머들은 **어셈블리 언어를 개발**하게 되었음

어셈블리 프로그램에서는 기호 이름과 CPU의 명령어가
일대일 대응되고 컴퓨터의 CPU가 달라지면 실행이 불가능함

저급 언어(Low Level Language)라고 불림

2) 프로그래밍 언어

어셈블리 언어

중간고사 성적과 기말고사 성적을 더하여 성적 합계를 구하는 연산을 어셈블리어로 표시한 예

```
MOV     AX,     MIDSCORE
MOV     CX,     FINALSCORE
ADD     AX      CX
```