

컴퓨터프로그래밍의기초

CLASS 1.

CHAPTER 01 프로그래밍의 개념



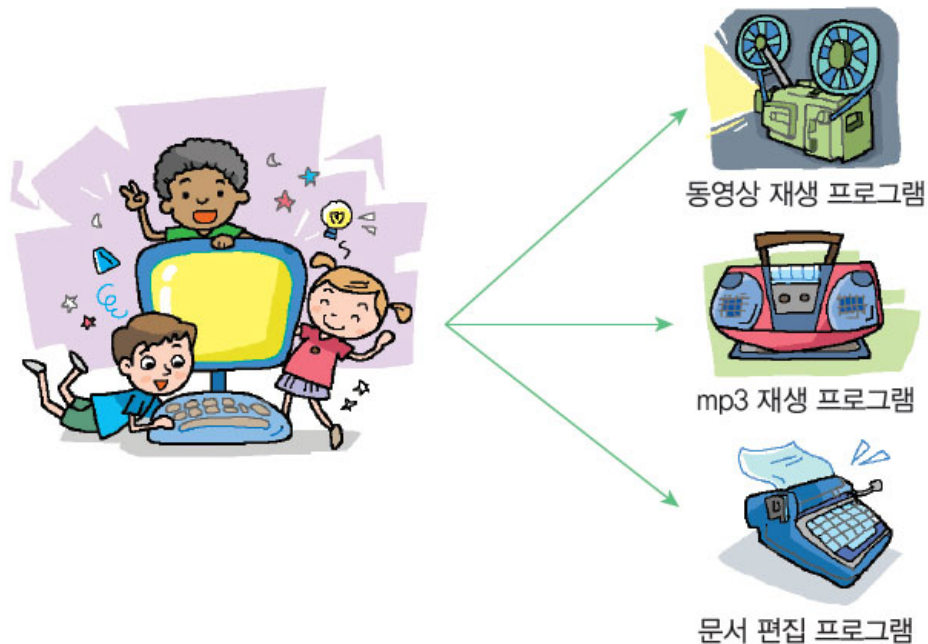
이번 장에서 학습할 내용



- 프로그래밍 개념
- 프로그래밍 언어
- 알고리즘 - 순서도

프로그램이란? == Software, Applicatoin

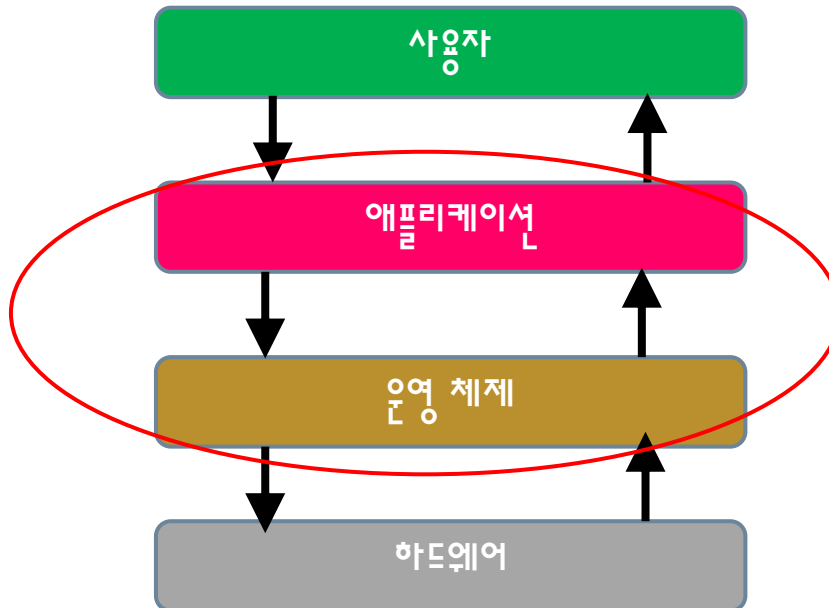
- 프로그램, Software(SW), Application(App)
- 컴퓨터는 범용적인(여러 분야나 용도로 널리 쓰는) 기계이다.
 - ▣ -> 컴퓨터를 사용하여 많은, 다양한, 여러 작업을 할 수 있다.
- 컴퓨터를 범용적으로 만드는 것은 바로 프로그램이라는 개념을 사용하기 때문이다.



만약 컴퓨터에 프로그램이 없다면?

- 컴퓨터 하드웨어만 있고 프로그램이 없다면
컴퓨터는 그저 약간의 열과 소음을 발생하는 쓸모없는 기계에 불과하다.
- “윈도우즈”와 같은 운영 체제를 설치하고
추가로 여러가지 응용 프로그램을 설치해야만 비로소 유용하게 된다.

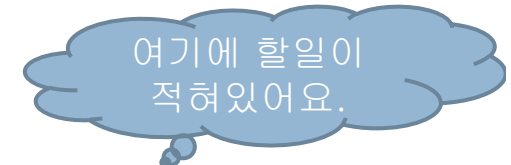
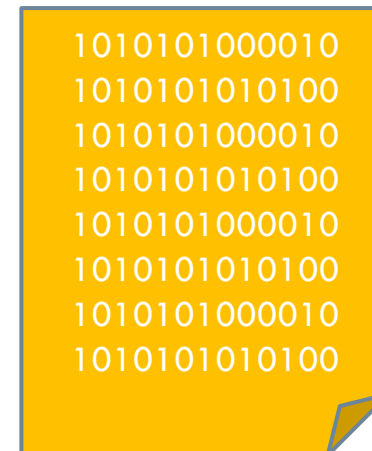
***** 소프트웨어(Software), Application, 프로그램, 응용 프로그램



프로그램은 설치하도록 했을까?

Q) 왜 컴퓨터에서는 가전제품처럼 프로그램 설치 없이 바로 동작되도록 하지 않고 불편하게 사용자가 프로그램을 설치하게 하였을까?

A) 컴퓨터를 범용적인 기계로 만들기 위해서이다.
컴퓨터는 프로그램만 바꾸어주면 다양한 작업을 할 수 있다.



계산기와 컴퓨터의 차이

정해진 기능만을 수행해요.
기능을 변경할 수는 없어요.

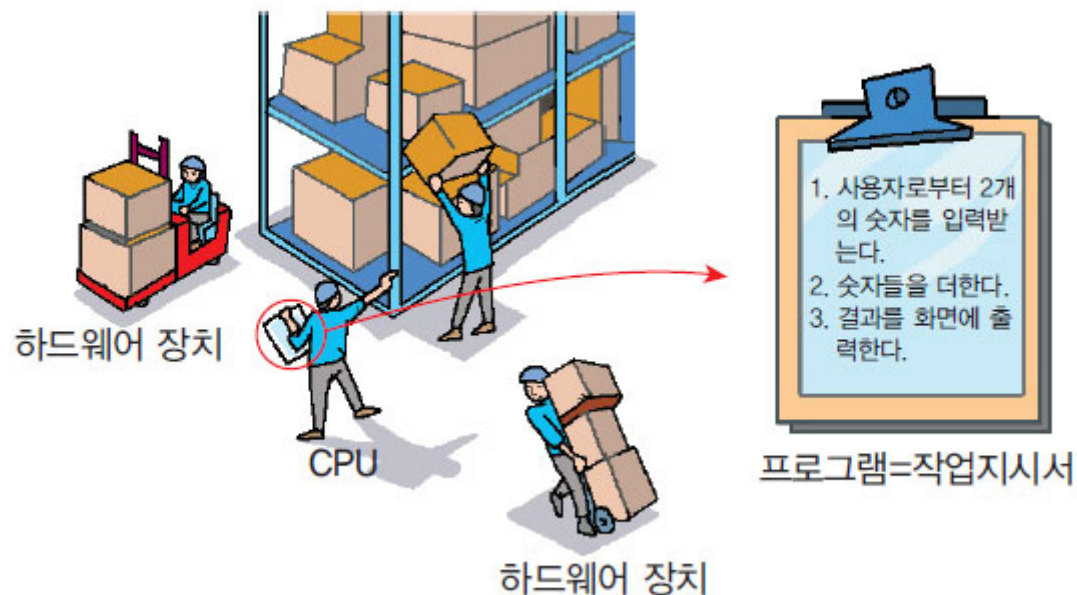


수행하는 기능을 바꿀 수
있다.



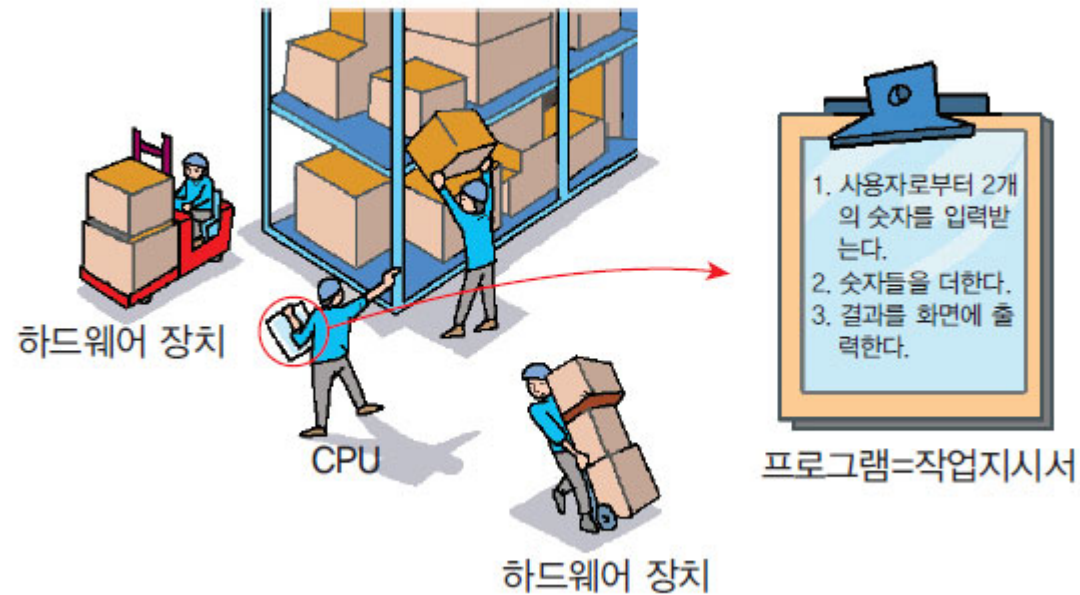
컴퓨터의 정의 *****

- 시스템: == 컴퓨터, HW + SW
- 컴퓨터(computer)는 단순히 계산(compute-CPU)만 하는 기계가 아니다.
- 컴퓨터는 프로그램(명령어-Instructions)들의 리스트)에 따라
- 데이터-data를 처리하는 기계라고 할 수 있다.



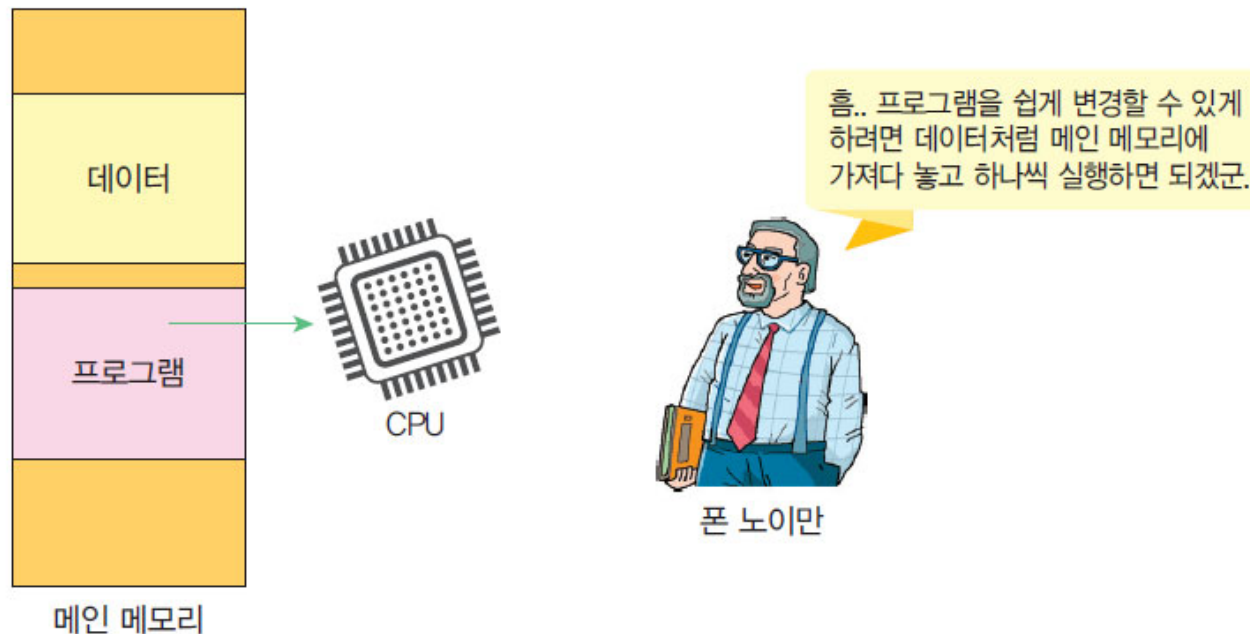
프로그램 안에는 무엇이 들어있을까?

- 프로그램은 특정한 작업을 위한 작업 지시서라고 보면 된다.
- 작업을 지시하려면 명령어(instruction)들을 나열해야 한다.
- 프로그램 안에는 명령어들이 들어 있다.



폰노이만 구조 *****

- 프로그램, **Software**는 메모리, 메인 메모리에 실행(**running**) 된다.
 - ▣ **Memory(RAM) -> (쉽게) 변경 가능!!**
- 메인 메모리에 로딩(**loading**)된 프로그램에서
- 명령어(**instructions**)들을 순차적으로 CPU로 가져와서 실행한다.
 - ▣ Bus, 버스: CPU와 메모리 사이에 1개의 버스 (명령과 데이터가 하나의 버스를 같이 사용.)
 - ▣ <-> 하버드 아키텍처(Harvard architecture): 명령용 버스와 데이터용 버스로 물리적으로 2개의 버스



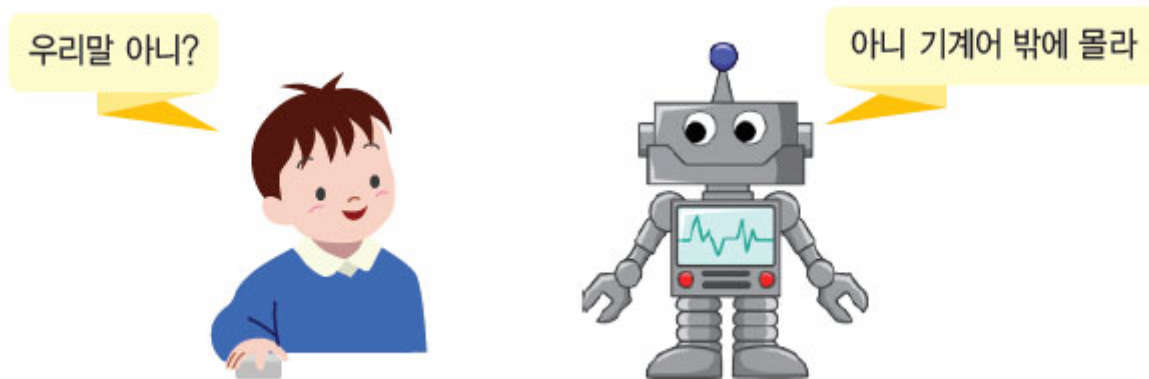
이번 장에서 학습할 내용



- 프로그래밍 개념
- 프로그래밍 언어
- 알고리즘

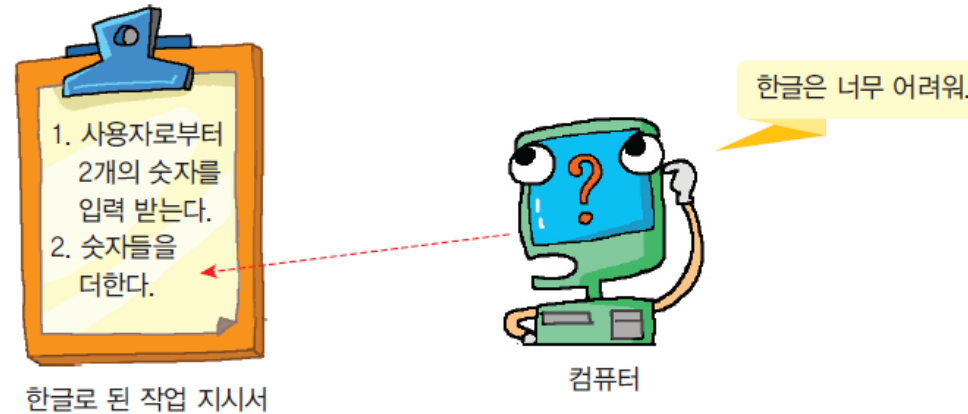
컴퓨터가 이해하는 언어 – 2진수 binary *****

- 컴퓨터가 알아듣는 언어는 한가지이다.
 - ▣ 전원버튼; 전기신호; 2진수! (binary)
- 0과 1로 구성되어 있는 “0011011101010...”과 같은 기계어이다.
- 컴퓨터는 모든 것을 0과 1로 표현하고
- 0과 1에 의하여 스위치 회로들이 ON/OFF 상태로 변경되면서 작업을 한다.



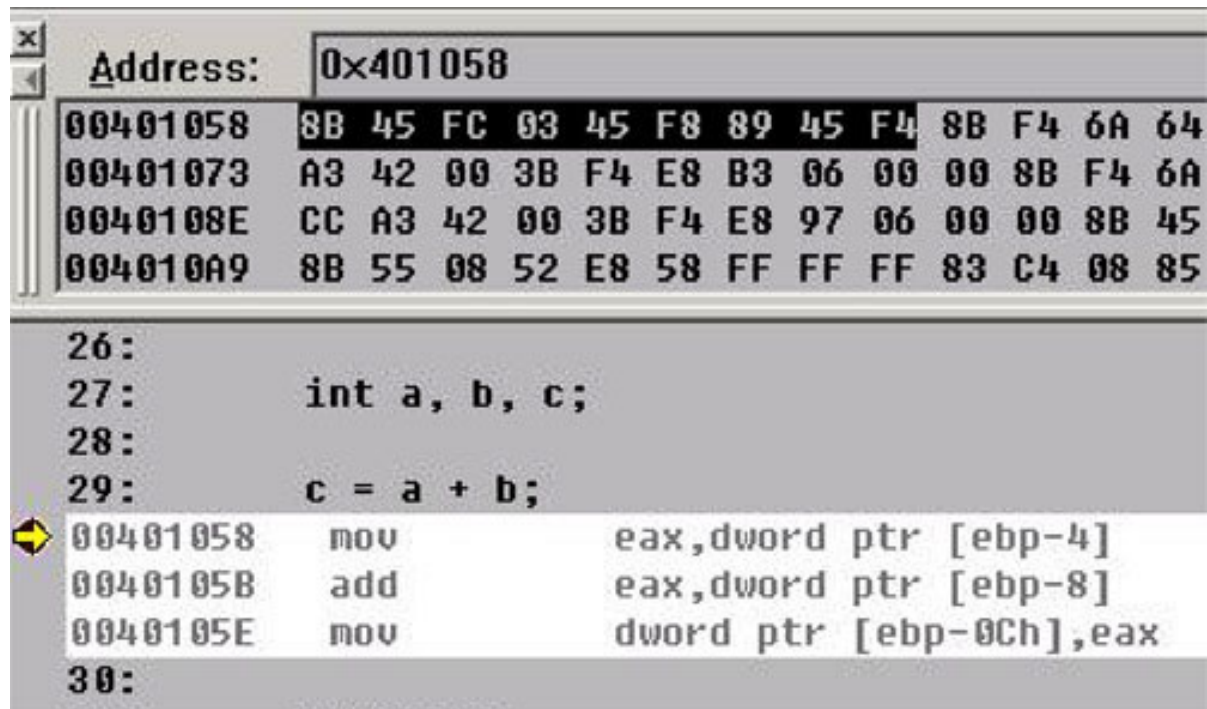
기계어(machine language)

- 0과 1로 구성되어 있는 “001101110001010...”과 같은 2진수이다.
- 비트 (bit)



기계어

□ 기계어의 예



The screenshot shows a debugger window with the address 0x401058. The assembly code is as follows:

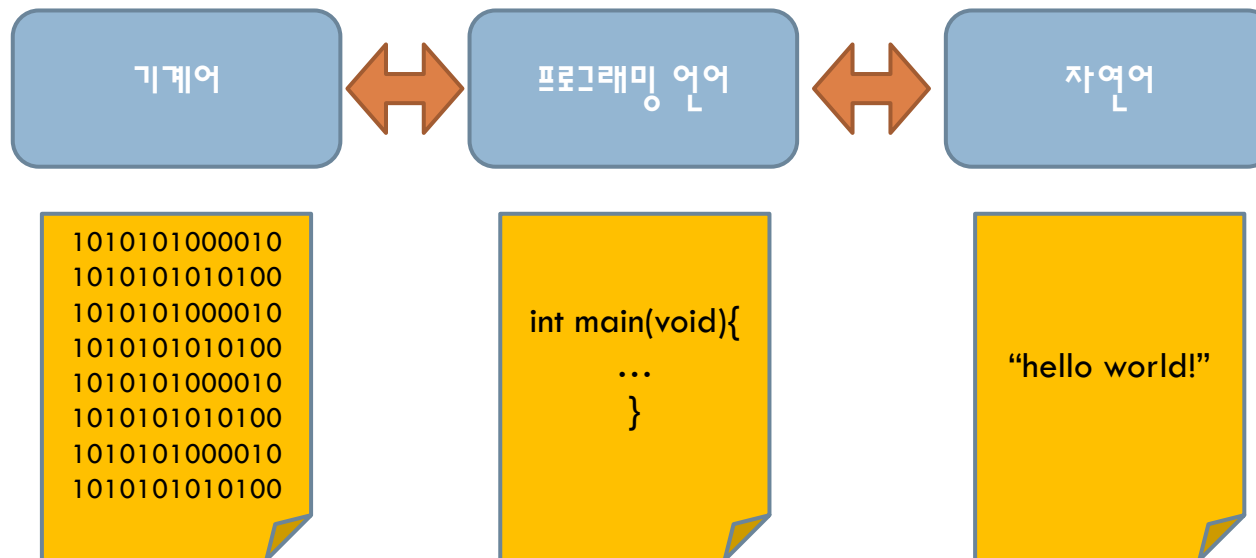
Address	Hex	Assembly
00401058	8B 45 FC 03 45 F8 89 45 F4	mov eax,dword ptr [ebp-4]
00401073	A3 42 00 3B F4 E8 B3 06 00 00 8B F4 6A	add eax,dword ptr [ebp-8]
0040108E	CC A3 42 00 3B F4 E8 97 06 00 00 8B 45	mov dword ptr [ebp-0Ch],eax
004010A9	8B 55 08 52 E8 58 FF FF FF 83 C4 08 85	

The assembly code is also displayed in a separate window below the hex view:

```
26:
27:     int a, b, c;
28:
29:     c = a + b;
30:
```

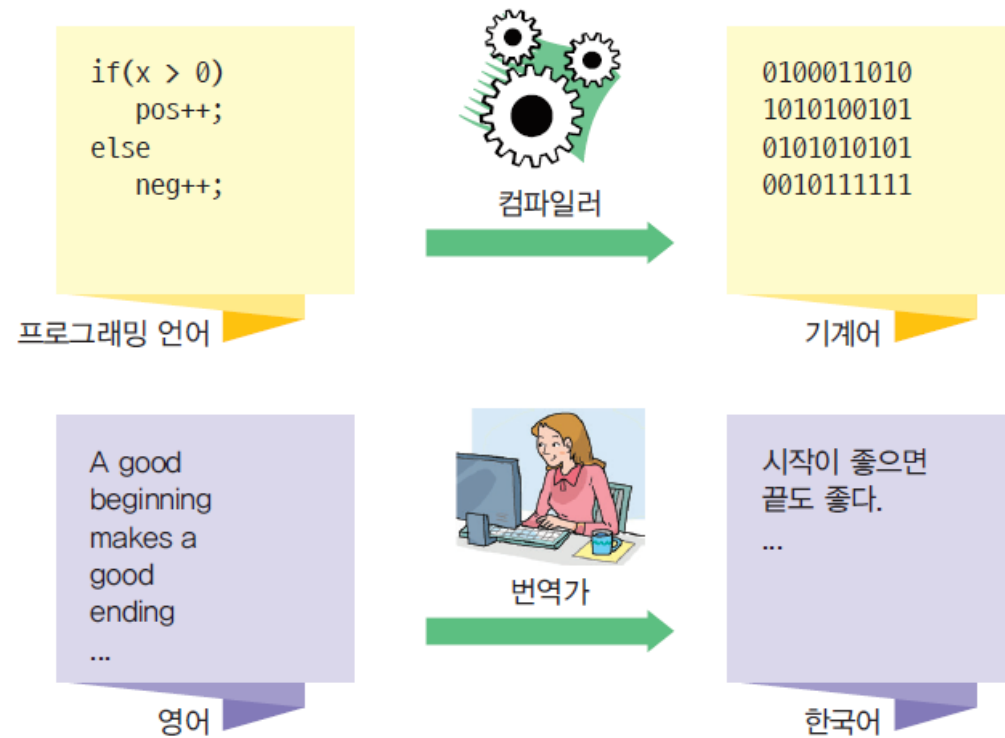
프로그래밍 언어 – C, C++, Java *****

- 기계어를 사용할 수는 있으나
 - 이진수로 프로그램을 작성하여야 하기 때문에 아주 불편하다.
- 프로그래밍 언어는 자연어와 기계어 중간 쯤에 위치
- **컴파일러가** 프로그래밍 언어를 기계어로 통역
 - **프로그래밍 언어 마다 배포되는 컴파일러가 있음 !!!**
 - **C → gcc: gnu compiler; GNU 컴파일러 모음 (GNU Compiler Collection)**



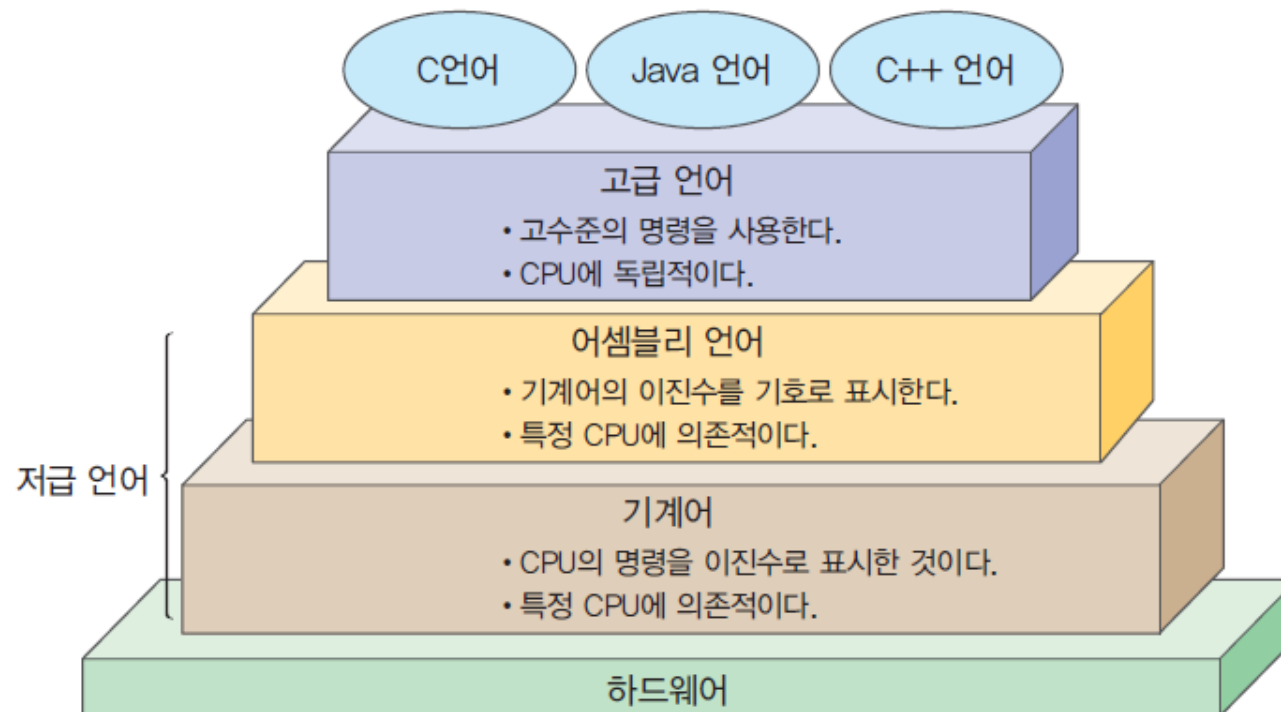
컴파일러

- 컴파일러(compiler)는 인간과 컴퓨터 사이의 통역이라 할 수 있다.



프로그래밍 언어의 분류

- 기계어(machine language)
- 어셈블리 언어(assembly language)
- 고급 언어(high-level language)



기계어

- 특정 컴퓨터의 명령어(instruction)를 이진수로 표시한 것
- 0과 1로 구성
- 하드웨어에 종속적이다. (Intel ; AMD ; ARM)

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

어셈블러

- 어셈블러 수업 → 하나의 언어, 학문
- **CPU**의 명령어들을 이진수가 아닌 영어의 약자인 기호로 표기
- 기계어보다는 더 높은 수준에서 프로그램을 작성하는 것을 가능
- 기호와 **CPU**의 명령어가 일대일 대응
- 어셈블러(**assembler**): 기호를 이진수로 변환하는 프로그램

```
MOV AX, MIDSCORE  
MOV CX, FINALSORE  
ADD AX CX  
MOV TOTALSCORE, AX
```

고급언어

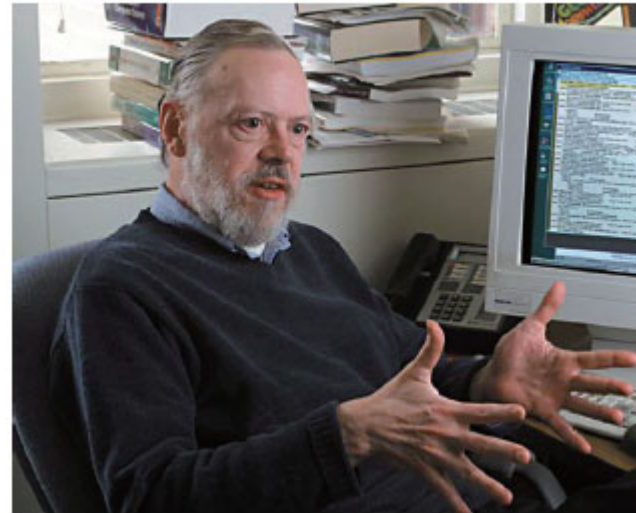
- 특정한 컴퓨터의 구조나 프로세서(CPU, HW)에 무관하게,
- 독립적으로 프로그램을 작성할 수 있는 언어
- C, C++, JAVA, FORTRAN, PASCAL
- 컴파일러: 고급 언어 문장을 기계어로 변환하는 프로그램

고급 언어의 종류

언어	특징	예제
FORTRAN	FORTRAN은 1950년대에 가장 먼저 만들어진 언어로서 수치 계산과 과학 기술 계산에 적합하다.	<pre> PROGRAM HELLO PRINT '(A)', 'Hello World' STOP END </pre>
COBOL	COBOL은 1959년 만들어진 비즈니스 사무 처리 언어이다. 이 언어는 구어체 문장 형태로 기술할 수 있도록 설계되었다.	<pre> IDENTIFICATION DIVISION. PROGRAM-ID. HELLO-WORLD. PROCEDURE DIVISION. DISPLAY 'Hello World'. STOP RUN. </pre>
Python	파이썬은 1991년 귀도 반 로섬(Guido van Rossum)이 개발한 인터프리트 언어이다. 초보자들이 배우기 쉬운 것으로 알려져 있다.	<pre> print("Hello World") </pre>
C	C언어는 1970년대 초반, UNIX 운영체제를 위하여 AT&T의 벨 연구소에서 일하던 데니스 리치에 의하여 만들어졌다.	<pre> int main(void) { printf("Hello World\n"); return 0; } </pre>
C++	C++는 1983년 벨 연구소의 스트로스트룹에 의하여 개발된 언어로서 C언어에 클래스 개념을 비롯하여 여러 가지 객체지향적인 특징들을 추가한 언어이다.	<pre> int main() { cout << "Hello World" << endl; return 0; } </pre>

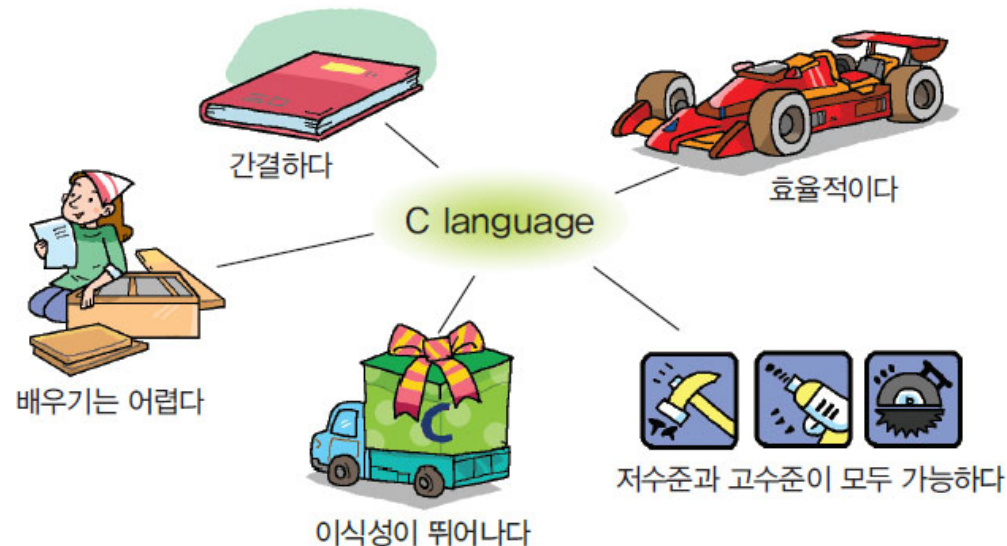
C언어의 소개

- 1970년대 초 AT&T의 **Dennis Ritchie** 에 의하여 개발
- UNIX 운영 체제 개발에 필요해서 만들어짐
- 처음부터 전문가용 언어로 출발



C언어의 특징

- 간결하다. 효율적이다.
- C 언어는 하드웨어를 직접 제어하는 하는 저수준의 프로그래밍도 가능하고 고수준의 프로그래밍도 가능하다.
- C언어는 이식성이 뛰어나다.
- C언어를 이해하면 컴퓨터 하드웨어가 어떻게 동작하는 지를 이해할 수 있다. *****



아무래도 C언어는 사용될 것인가?

- 최초의 언어. 부팅부터 운영체제까지 C언어로 구현.
- 임베디드 시스템에서 C언어가 대부분.
 - Embedded: SoC, 반도체, 칩 Chip

임베디드 시스템: == 컴퓨터

임베디드 시스템이란 컴퓨터가 특정 장비(핸드폰, 자동차) 안에 그게 맞게 설계 구현되서 들어 간 것.

→ 핸드폰, 가전, 자동차, 로봇

이번 장에서 학습할 내용



- 프로그램의 이해
- 프로그래밍 언어
- 알고리즘

알고리즘이란?

- 알고리즘(algorithm)
 - ▣ 문제를 풀기 위하여 컴퓨터가 수행하여야 할 단계적인 절차를 기술한 것
 - ▣ 단계적인 절차; 순서
- (예) 전화번호부에서 특정한 사람(박철수라고 가정하자)의 전화번호를 찾는 문제를 생각하여 보자.



1부터 10까지의 합을 구하는 알고리즘

① 1부터 10까지의 숫자를 직접 하나씩 더한다.

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

② 두수의 합이 10이 되도록 숫자들을 그룹핑하여 그룹의 개수에 10을 곱하고 남은 숫자 5를 더한다.

$$\begin{array}{l} (0 + 10) = 10 \\ (1 + 9) = 10 \\ (2 + 8) = 10 \\ (3 + 7) = 10 \\ (4 + 6) = 10 \\ 5 \end{array} \left. \vphantom{\begin{array}{l} (0 + 10) = 10 \\ (1 + 9) = 10 \\ (2 + 8) = 10 \\ (3 + 7) = 10 \\ (4 + 6) = 10 \end{array}} \right\} \begin{array}{c} 10 * 5 = 50 \\ + \\ 5 \\ = \\ 55 \end{array}$$



③ 공식을 이용하여 계산할 수도 있다.

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

알고리즘의 기술

- 자연어(natural language)
- 순서도(flowchart)
- 의사 코드(pseudo-code)



컴퓨터 앞에 바로
앉지 말고 알고리즘을
구상하여야 합니다.

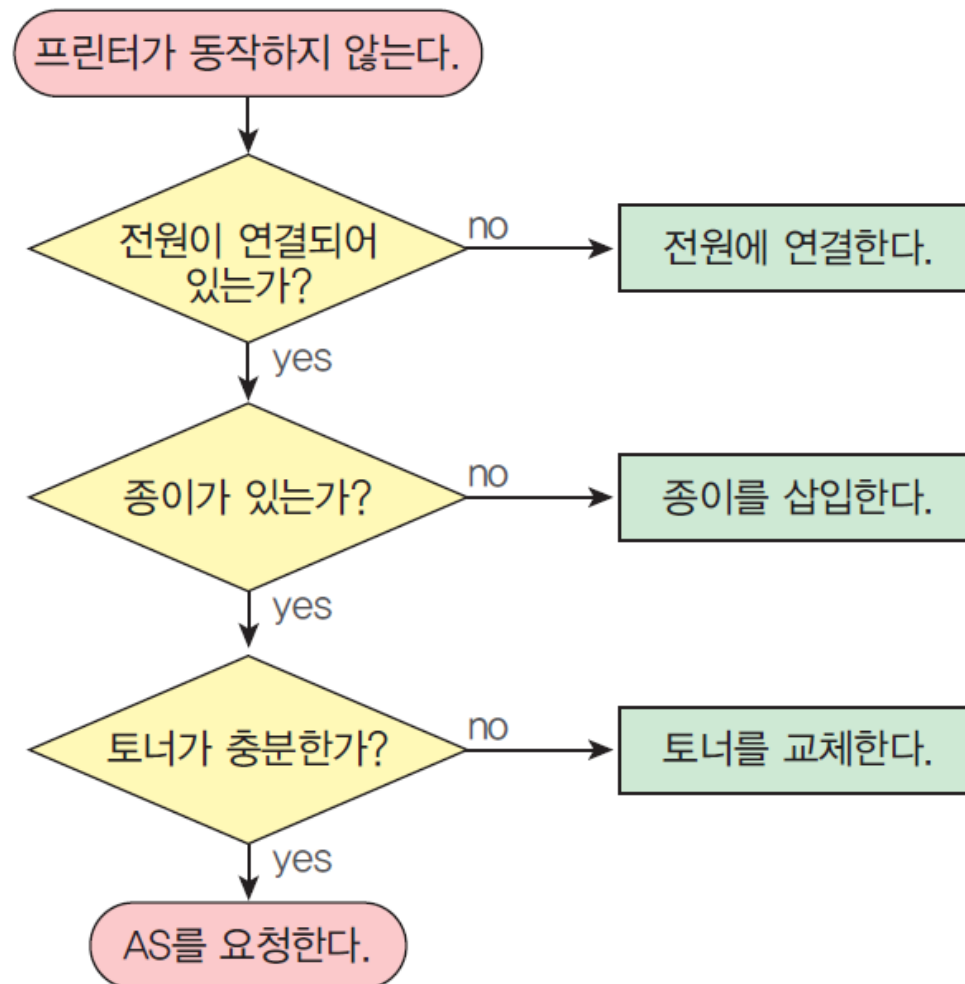
알고리즘의 기술

□ 순서도(flow chart) *****

- ▣ 프로그램에서의 논리 순서 또는 작업 순서를 그림으로 표현하는 방법



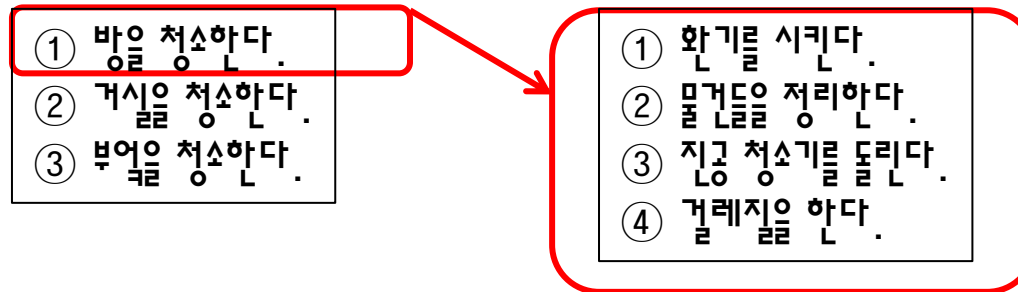
알고리즘의 예



알고리즘을 만드는 방법

- 분석과 세분화

1. 문제를 한 번에 해결하려고 하지 말고 더 작은 크기의 문제들로 분해한다.
2. 문제가 충분히 작아질 때까지 계속해서 분해한다.



환기



물건정리



진공청소기

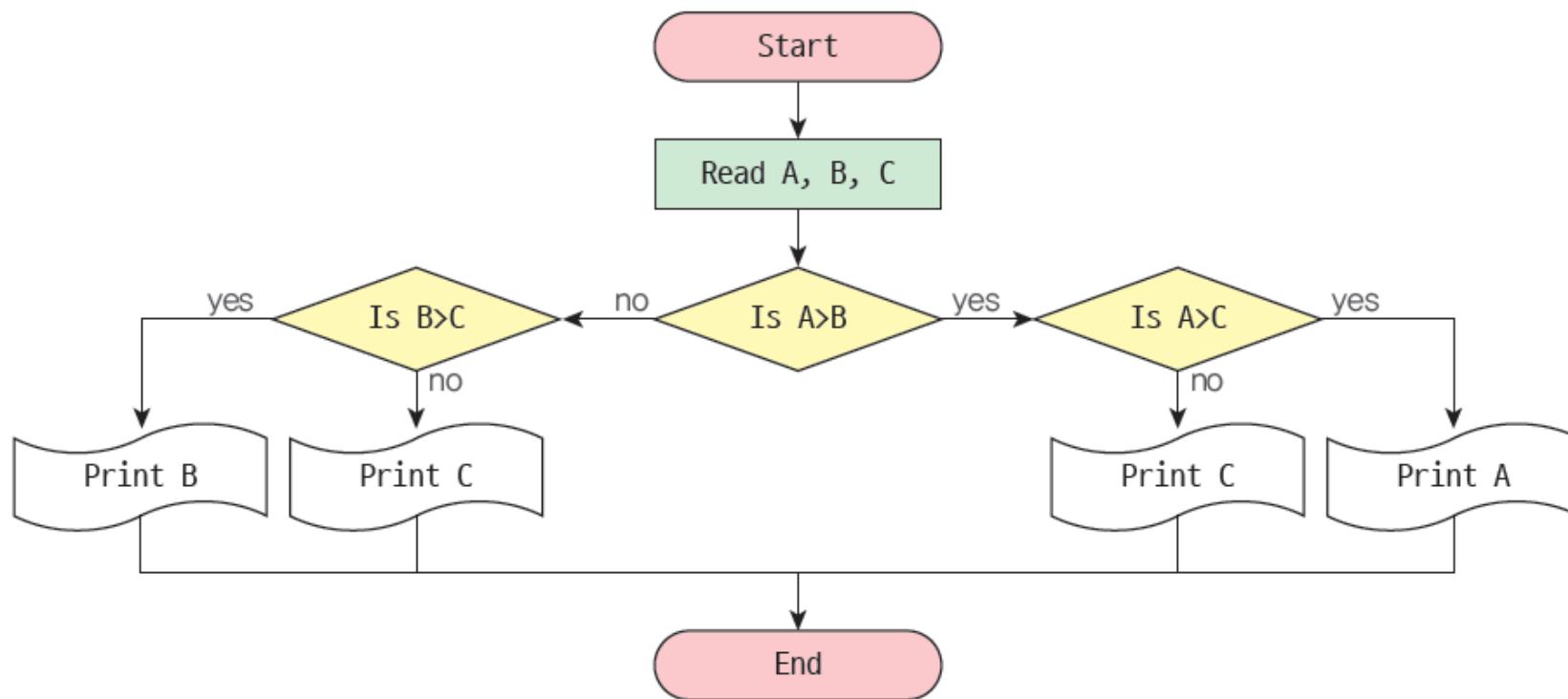


걸레질

Mini Project: 3개의 수 중에서 최대값 찾기

- 사용자로부터 받은 3개의 수 중에서 최대값을 찾는 알고리즘을 순서도로 작성해보자.
- 1. 사용자로부터 받은 3개의 수를 A, B, C라고 하자.
- 2. 먼저 A와 B를 비교한다. A가 B보다 크면 A와 C를 비교해서 큰 수를 출력하면 된다.
- 3. 만약 B가 A보다 크다면 B와 C를 비교하여서 큰 수를 출력하면 된다.

알고리즘



Q & A



- In conclusion,
 - 프로그래밍 개념
 - C언어 개념
 - 알고리즘 - 순서도