Introduction to Computers and Programming

서울대학교 전기컴퓨터공학부
홍 성수 교수
sshong@redwood.snu.ac.kr
http://redwood.snu.ac.kr

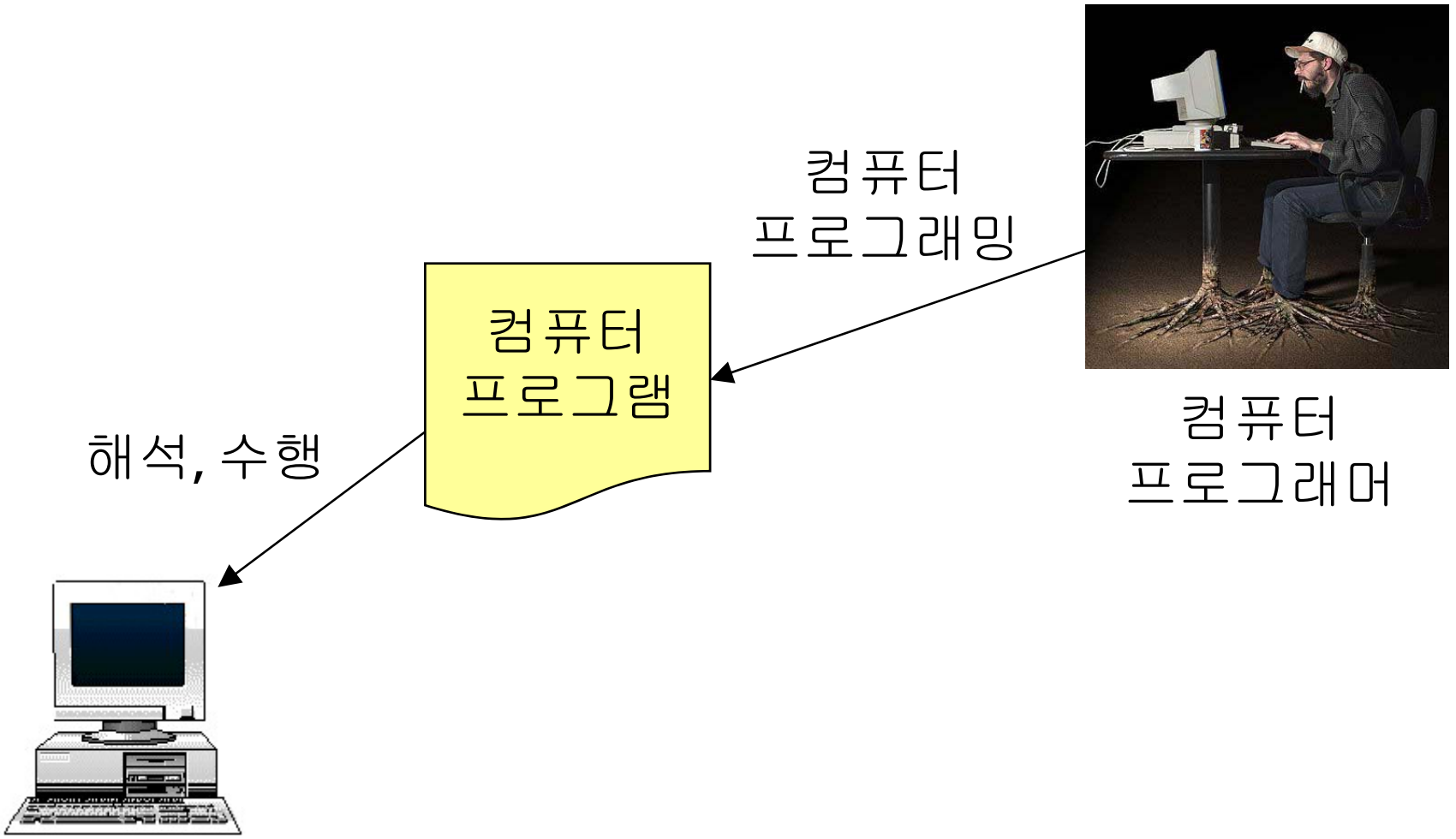
Topic 0: 컴퓨터와 프로그래밍

프로그래밍

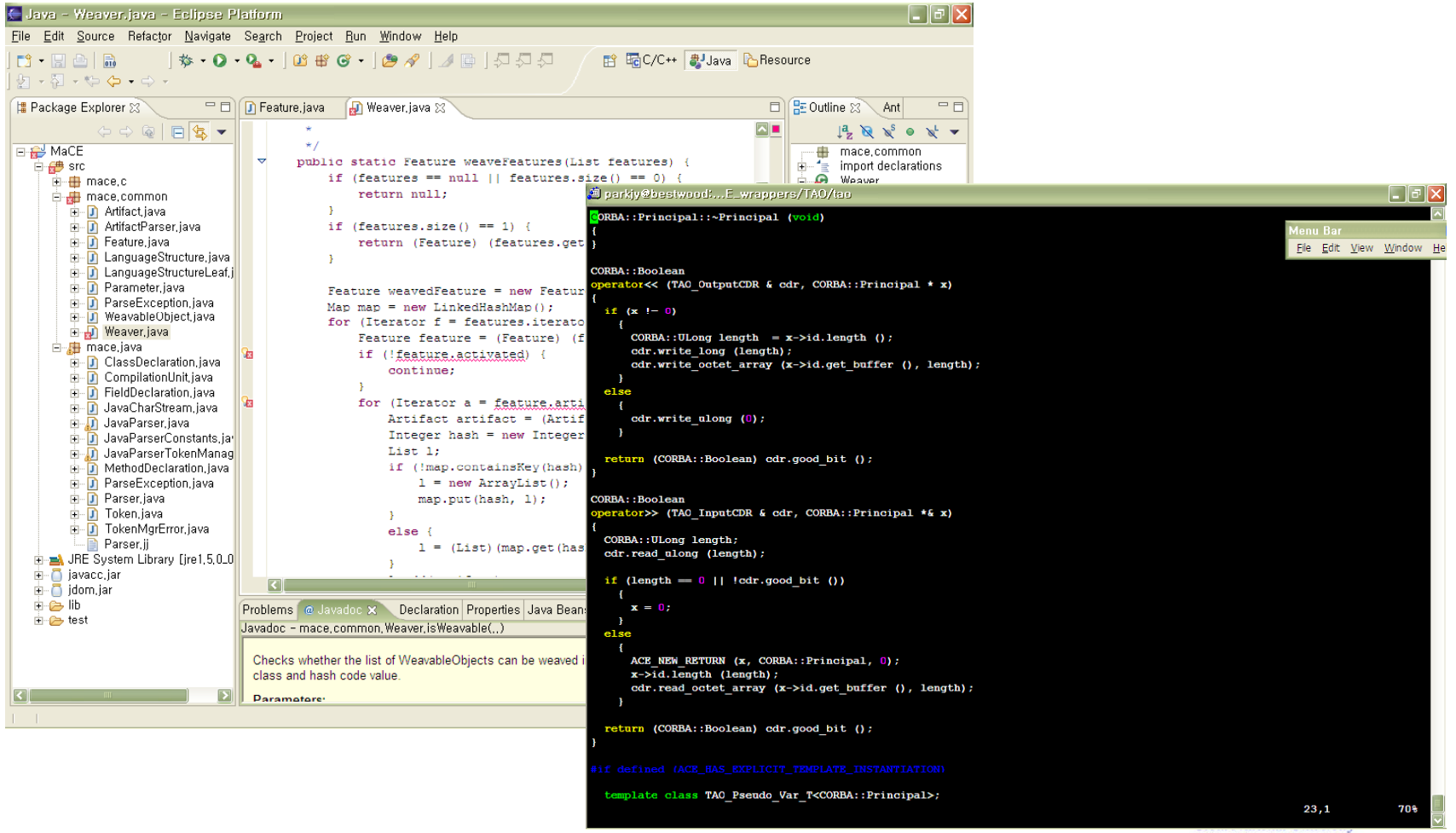
프로그래밍이란?

- 프로그래밍:
  - The planning, scheduling, or performing of a task or an event.
- 컴퓨터 프로그래밍:
  - The process of planning a sequence of instructions for a computer to follow.
- 컴퓨터 프로그램:
  - A sequence of instructions outlining steps to be performed by a computer.

프로그래밍이란?



컴퓨터 프로그래밍의 예


Java - Weaver.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Feature.java
Weaver.java
Outline
Ant
MaCE
src
mace.c
mace.common
Artifact.java
ArtifactParser.java
Feature.java
LanguageStructure.java
LanguageStructureLeaf.j
Parameter.java
ParseException.java
WeavableObject.java
Weaver.java
mace.java
ClassDeclaration.java
CompilationUnit.java
FieldDeclaration.java
JavaCharStream.java
JavaParser.java
JavaParserConstants.ja
JavaParserTokenManag
MethodDeclaration.java
ParseException.java
Parser.java
Token.java
TokenMgrError.java
Parser.jj
JRE System Library [jre1.5.0_0
javacc.jar
jdom.jar
lib
test
public static Feature weaveFeatures(List features) {
if (features == null || features.size() == 0) {
return null;
}
if (features.size() == 1) {
return (Feature) (features.get
}
Feature weavedFeature = new Featur
Map map = new LinkedHashMap();
for (Iterator f = features.iterato
Feature feature = (Feature) (f
if (!feature.activated) {
continue;
}
for (Iterator a = feature.arti
Artifact artifact = (Artif
Integer hash = new Integer
List l;
if (!map.containsKey(hash)
l = new ArrayList();
map.put(hash, l);
}
else {
l = (List)(map.get(has
}
Problems
Javadoc
Declaration
Properties
Java Bean
Javadoc - mace.common.Weaver.isWeavable(...)
Checks whether the list of WeavableObjects can be weaved i
class and hash code value.
Parameters:
mace.common
import declarations
Weaver
parkjy@bestwood:...E_wrappers/TAO/tao
CORBA::Principal::~Principal (void)
{
}
CORBA::Boolean
operator<< (TAO_OutputCDR & cdr, CORBA::Principal * x)
{
if (x != 0)
{
CORBA::ULong length = x->id.length ();
cdr.write_long (length);
cdr.write_octet_array (x->id.get_buffer (), length);
}
else
{
cdr.write_ulong (0);
}
return (CORBA::Boolean) cdr.good_bit ();
}
CORBA::Boolean
operator>> (TAO_InputCDR & cdr, CORBA::Principal *& x)
{
CORBA::ULong length;
cdr.read_ulong (length);
if (length == 0 || !cdr.good_bit ())
{
x = 0;
}
else
{
ACE_NEW_RETURN (x, CORBA::Principal, 0);
x->id.length (length);
cdr.read_octet_array (x->id.get_buffer (), length);
}
return (CORBA::Boolean) cdr.good_bit ();
}
#if defined (ACE_HAS_EXPLICIT_TEMPLATE_INSTANTIATION)
template class TAO_Pseudo_Var_T<CORBA::Principal>;
23,1
70%
Menu Bar
File Edit View Window He

언제 컴퓨터 프로그래밍을 하는가?

- 컴퓨터를 이용하여 문제를 풀고 싶을 때
  - 문제를 푸는 방법은 알고 있으나 계산량이 많거나 복잡할 때
  - 예) 2차 방정식, 최단 경로 찾기, 체스, ...
- 컴퓨터를 이용하여 하드웨어 장치들을 조작하고 싶을 때
  - 예 1) 키보드에서의 입력, 모니터로의 출력
  - 예 2) 마이크로 마우스, 자동차, 비행기, ...
- 대부분의 프로그램은 이 두 가지 경우를 함께 다룬다.
  - 예) 마이크로 마우스의 컴퓨터 프로그램은 미로 찾기 문제를 풀어야 한다. 또한, 센서 데이터를 읽고, 모터를 구동하여 마이크로 마우스를 움직여야 한다.

프로그래밍의 단계 (1)

문제 해결 단계 (Problem-Solving Phase)

1. 분석 (analysis):
   - Understand (define) the problem.
2. 범용 방법론 수립 (general solution or algorithm):
   - Develop a logical sequence of steps to be used to solve the problem.
3. 검사 (test):
   - Follow the exact steps as outlined to see if the solution truly solves the problem.

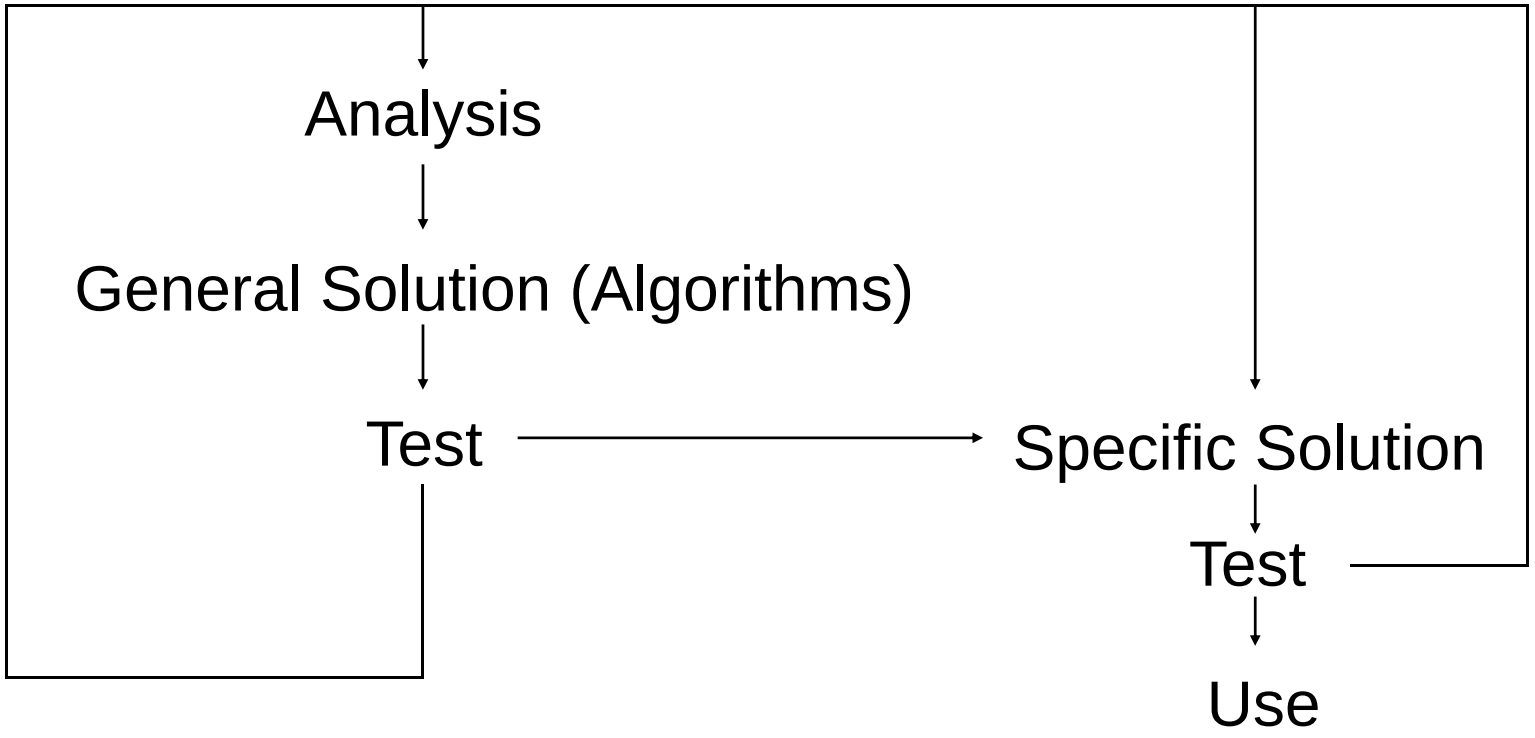
프로그래밍의 단계 (2)

구현 단계 (Implementation Phase)

4. 구체적 방법의 수립 (specific solution or program):
   - Translate the algorithm into a programming language (code).
5. 검사 방법의 수립 (test):
   - Have the computer follow the instructions.
   - Check the results and make corrections until the answers are correct.
6. 사용 (use):
   - Use the program.

프로그래밍의 단계 (3)

Problem-Solving Phase **Implementation Phase**



문제 해결을 위한 프로그래밍의 절차

문제: 2차 방정식을 풀어라.

1. 분석:
   주어진 실수 a, b, c가 있을 때 방정식 $ax^2 + bx + c = 0$ 을 만족하는 x의 값들을 구한다.

2. 범용 방법론 수립:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

한 문제에 대한 범용 방법론은 여러 가지가 있을 수 있다.

3. 검사:
   x를 방정식에 대입
   옳음을 "증명"

허근일 경우에는?

문제 해결을 위한 프로그래밍의 절차

4. 코딩 (프로그램 작성)

```
double a, b, c;
double temp1, temp2, temp3;
double x1, x2;
temp1 = -b;
temp2 = sqrt(b*b - 4*a*c);
temp3 = 2*a;
x1 = (temp1 + temp2) / temp3;
x2 = (temp1 - temp2) / temp3;
```

한 문제에 대한 프로그램은 여러 가지가 있을 수 있다.

5. 검사

컴퓨터에 프로그램을 설치한 다음 수행시키고, 다양한 조합의 a, b, c를 이용하여 결과 값이 올바른지 검사한다.

6. 사용

알고리즘과 프로그램 (1)

- 알고리즘 (algorithm):
  - A step-by-step procedure for solving a problem in a finite amount of time.
- 프로그래밍 언어:
  - A set of rules, symbols, and special words used to construct a program.
  - Is essentially an unambiguous and simplified form of English (with math symbols) that adheres to a set of grammar rules.
  - Programming requires you to develop a skill for writing very simple and exact instructions.
- 코딩:
  - Translating an algorithm into a programming language.

알고리즘과 프로그램 (2)

- 구현 (implementation):
  - The combination of coding and testing an algorithm.

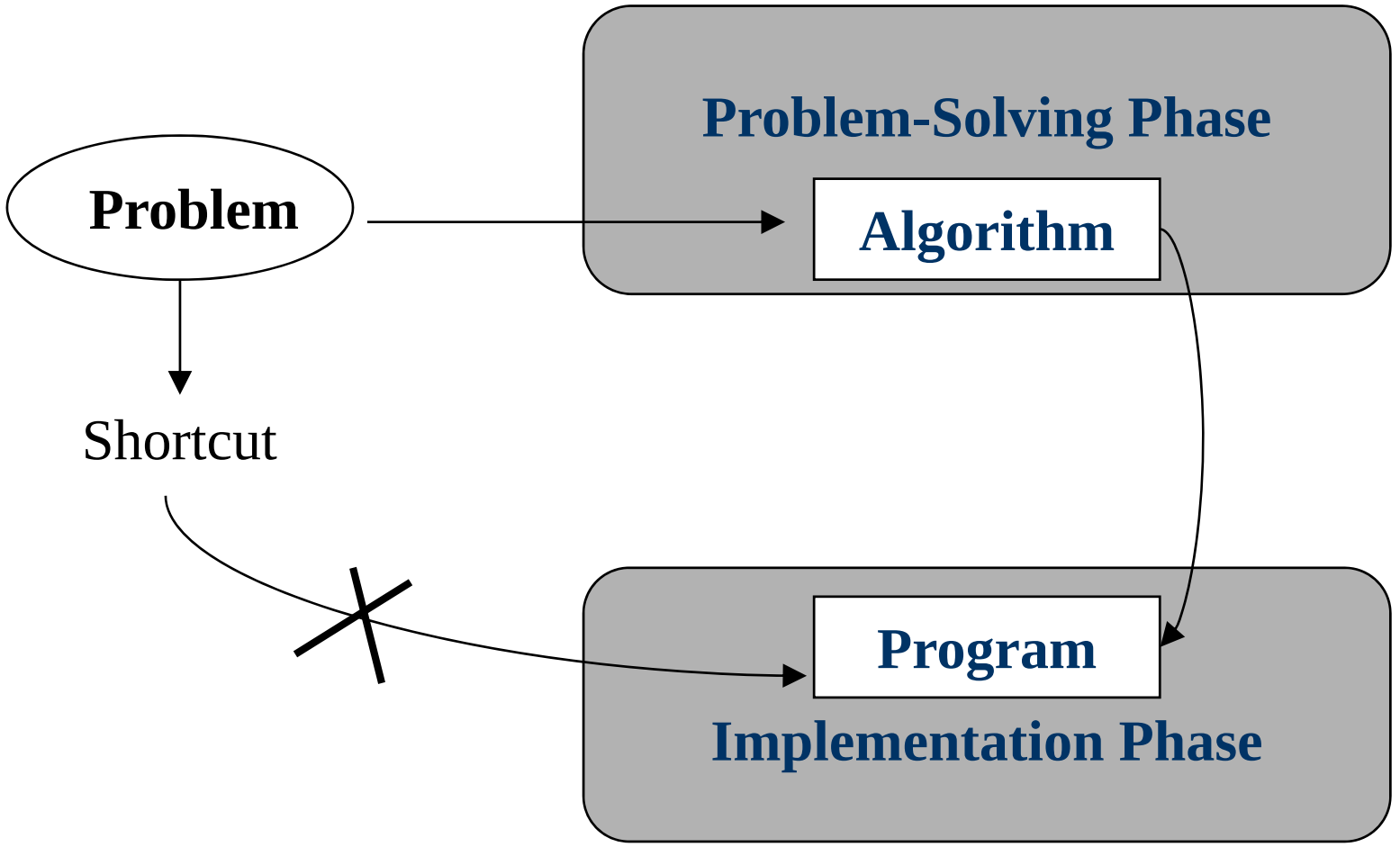
알고리즘과 프로그램 (3)

- 프로그램:
  - An algorithm expressed in a programming language.
- 프로그래밍은 단순히 프로그램 작성 이상의 의미를 갖는다.
  - A programmer must analyze the problem in order to develop a strategy that can be used in the program to solve the specific problem correctly.
  - Developing a general solution before actually writing the program helps the programmer manage the problem, keep thoughts straight, and avoid unnecessary errors.
  - Since most programs will be used over and over, program documentation and maintenance are important parts of programming.

소프트웨어 엔지니어링

- 문서화 (documentation):
  - Written text and comments that make a program easier for others to understand, use, and modify.
- 유지 보수 (maintenance):
  - The modification of a program, after it has been completed, in order to meet changing requirements or take care of any errors that show up.
  - 일반 휴대폰 → 카메라 폰의 등장

올바른 프로그래밍 습관


Problem-Solving Phase
Problem
Algorithm
Shortcut
Program
Implementation Phase

컴퓨터

데이터의 표현기법

- 이진수 시스템 (binary number system):
  - 정보가 '0'과 '1'의 조합으로 표현됨.
- 코딩:
  - The process of assigning bit patterns to pieces of information.
  - In the early days of computing, programming meant translating an algorithm into bit patterns.
- 정보 (information):
  - Any knowledge that can be communicated.
- 데이터:
  - Information that has been put into a form usable by a computer, that is, a form suitable for analysis or decision making.

정보 v.s. 데이터

정보	데이터	이진수 표현
온도가 9도 이다.	9	0000001001
A 키가 눌렸다.	A	010010
키가 178.2 cm 이다.	178.2	1001110
막다른 골목이다.	True	1

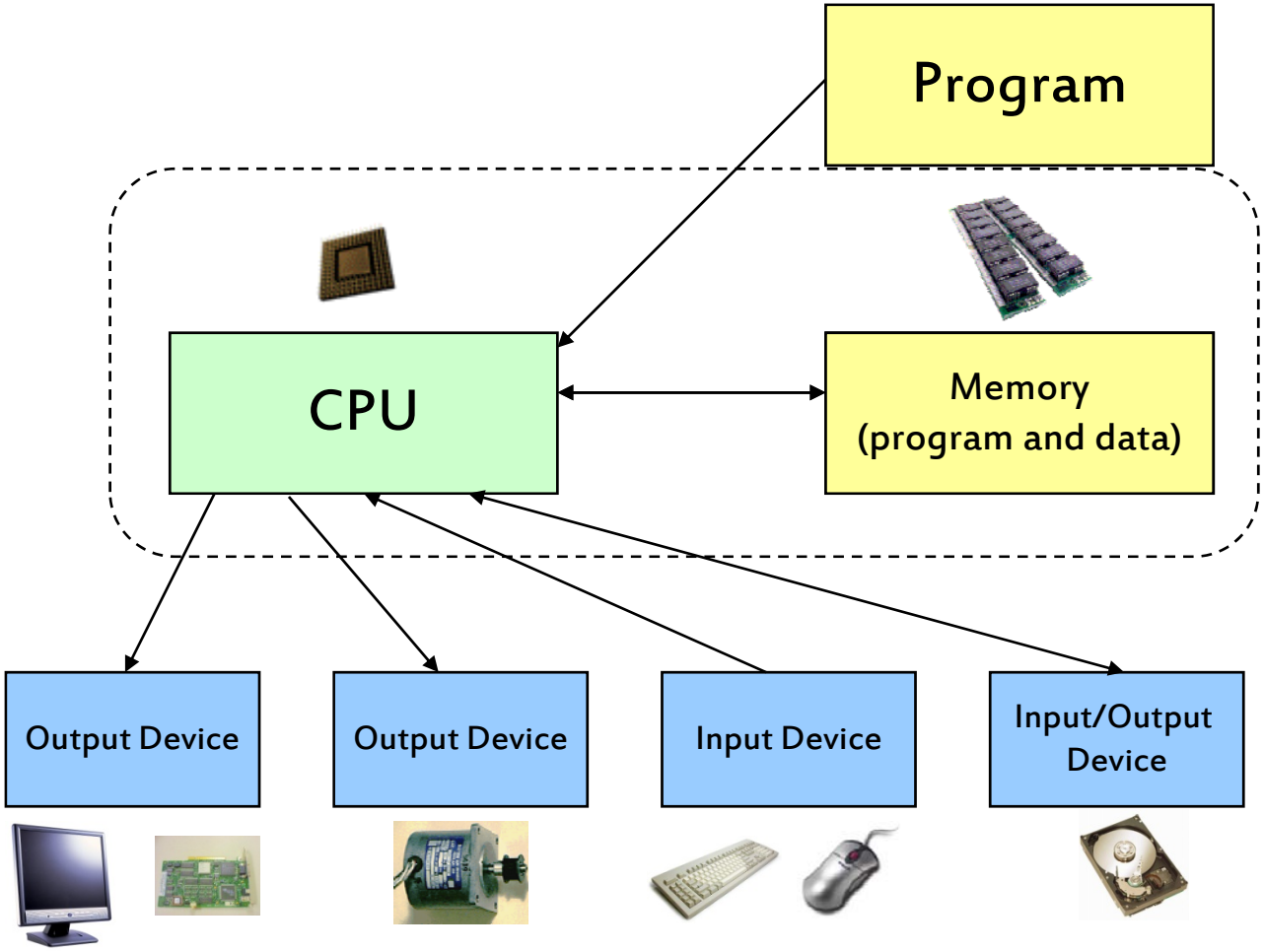
컴퓨터란? (1)

- 컴퓨터:
  - A programmable electronic device that can store, retrieve, and process data.
  - Consists of the five basic components:
    - memory, arithmetic/logic, control, input, and output.
- 메모리:
  - The internal data storage of a computer.
- 중앙처리장치 (central processing unit, microprocessor unit):
  - The part of the computer that executes the instructions of a program stored in memory;
  - Composed of the arithmetic/logic and the control units.

컴퓨터란? (2)

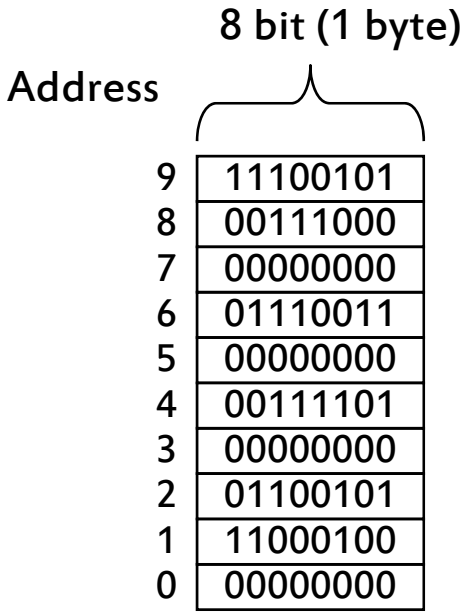
- 산술/논리 연산 장치 (ALU):
  - The computer component that performs arithmetic operations (addition, subtraction, multiplication, division) and logical operations (comparison of two values).
- 제어 장치:
  - The computer component that controls the actions of other components in order to execute instructions (the program) in sequence.
- 입출력 장치:
  - The parts of a computer that accept data to be processed (input) and/or present the results of the processing (output).

컴퓨터란


Program
CPU
Memory
(program and data)
Output Device
Output Device
Input Device
Input/Output
Device

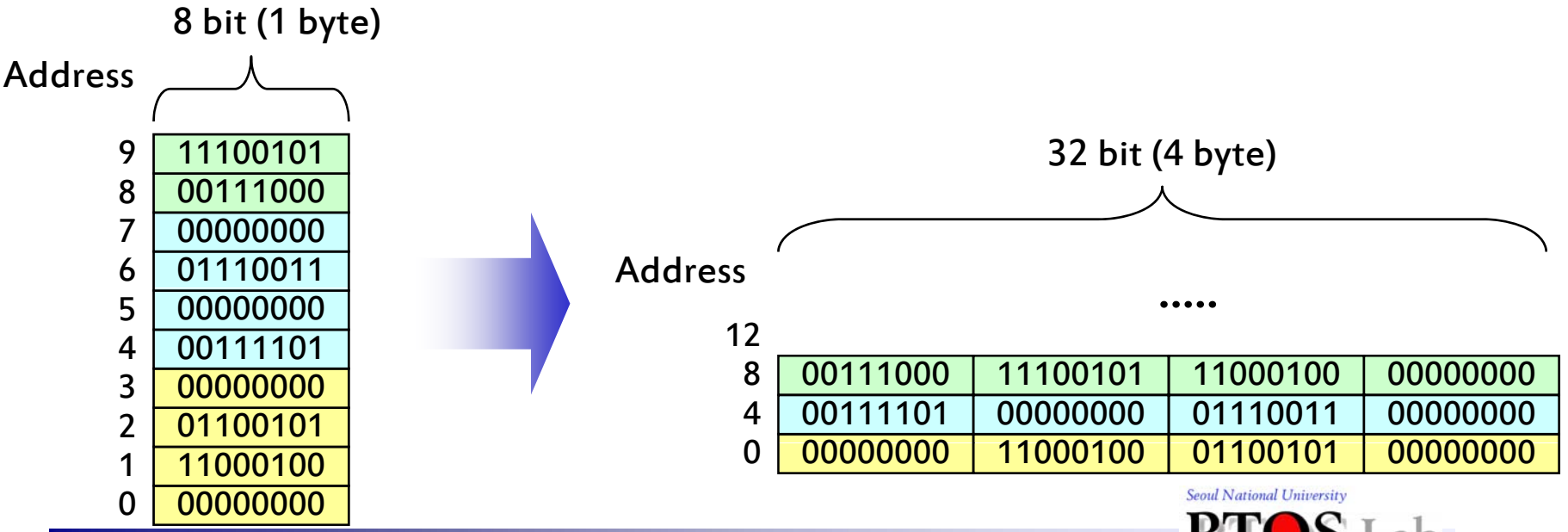
Memory

- Memory is an array of binary data.
- Each element in the array has address.
- Addresses are assigned for each 8 bit data.



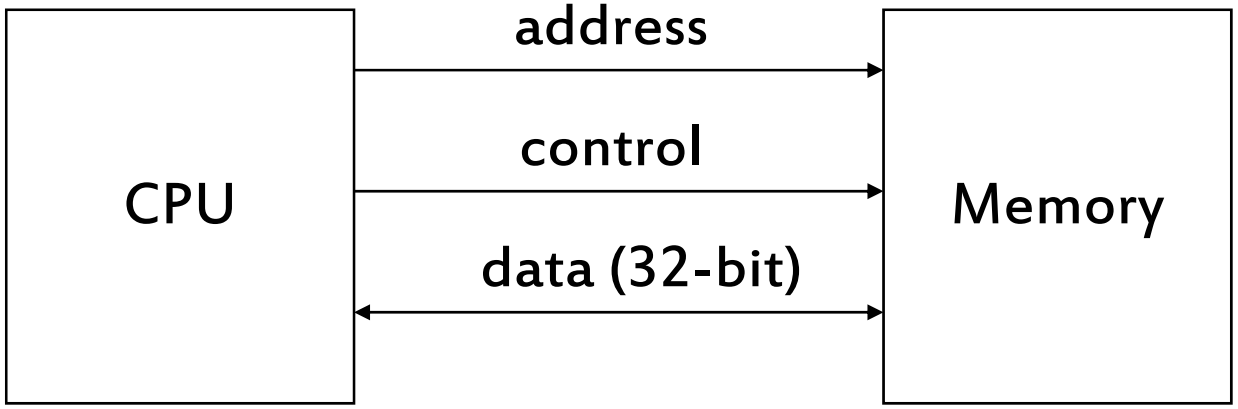
Memory

- Actually, CPU uses consecutive four bytes (32-bit) data to represent larger number.
  - 8 bit: 256 kinds of number
  - 32 bit: 4,294,967,296 kinds of number



CPU and Memory

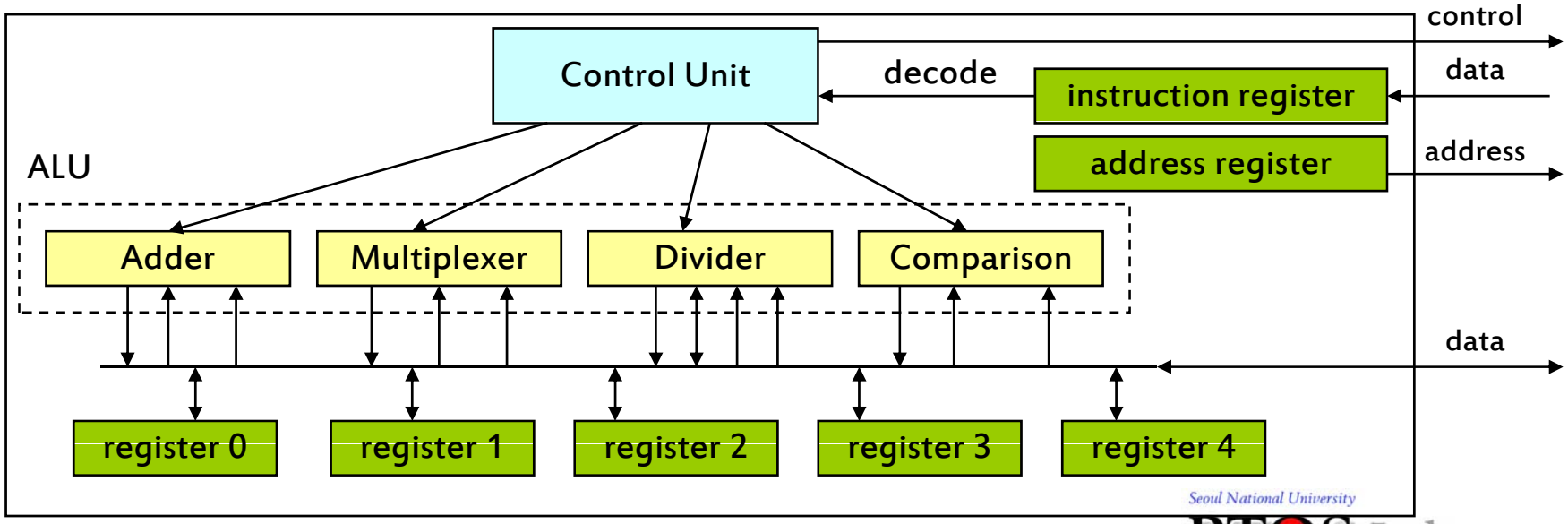
- Memory unit has three groups of signals
  - Address
  - Control: whether read or write
  - Data



CPU

- CPU consists of three components
  - Arithmetic/Logic Unit (ALU): 계산, 비교 수행
  - Control Unit: 명령어를 해석하여 ALU를 작동시킴
  - Registers: 데이터나 명령어를 임시 저장

CPU



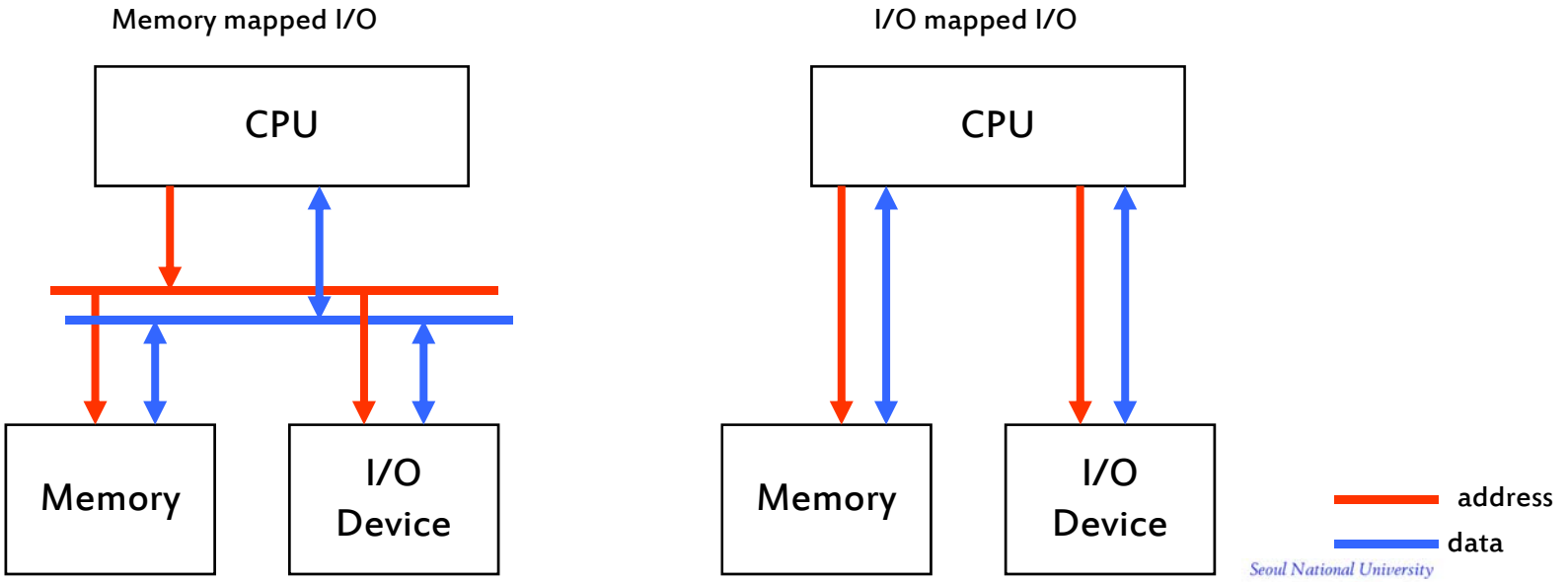
I/O Device

- I/O device also consists of three groups of signals
  - Address
  - Data
  - Control

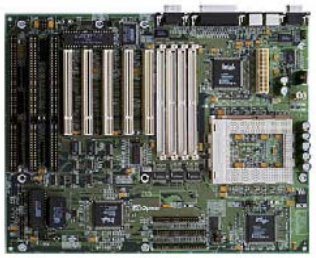
- 그러나

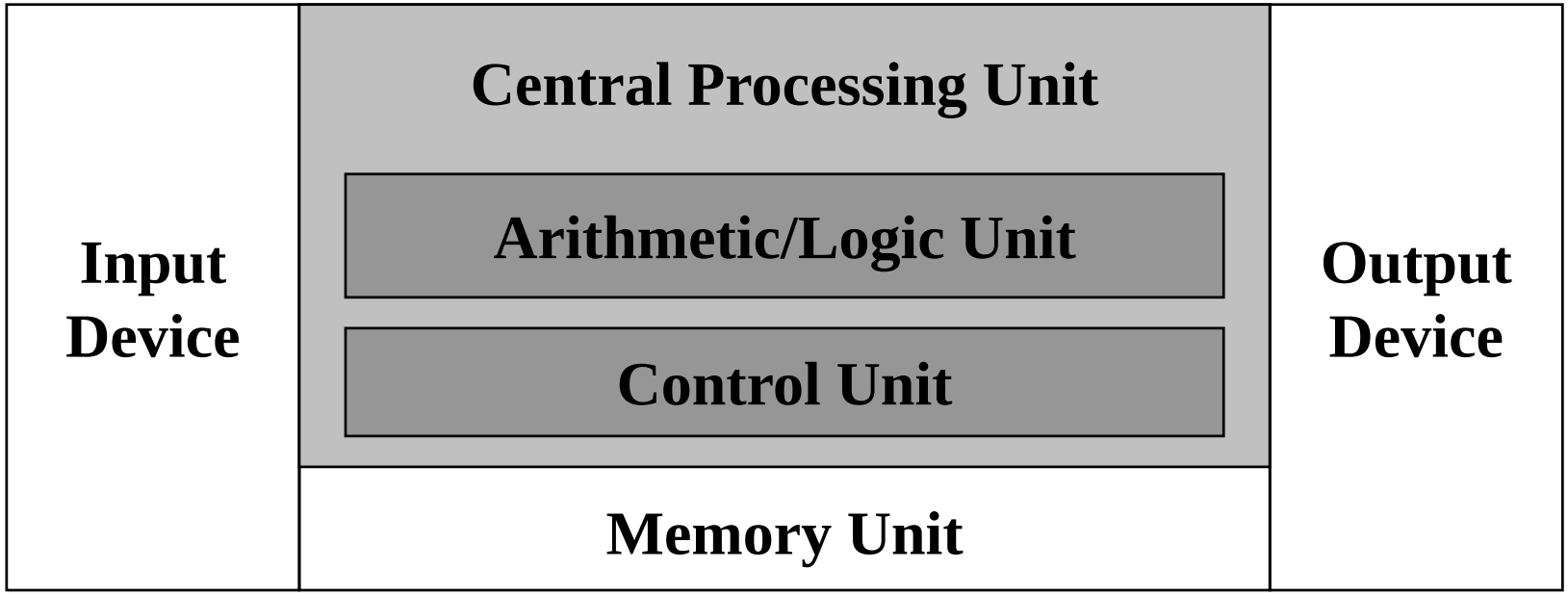
I/O Device

- Two kinds of ways to control I/O device
  - Memory mapped I/O: share address and data lines with memory unit
  - I/O mapped I/O: separate address and data lines



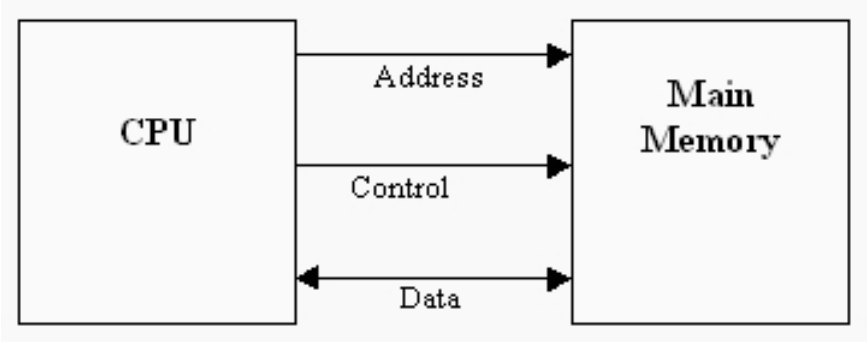
컴퓨터의 기본 구조


Central Processing Unit
Arithmetic/Logic Unit
Control Unit
Input Device
Output Device
Memory Unit

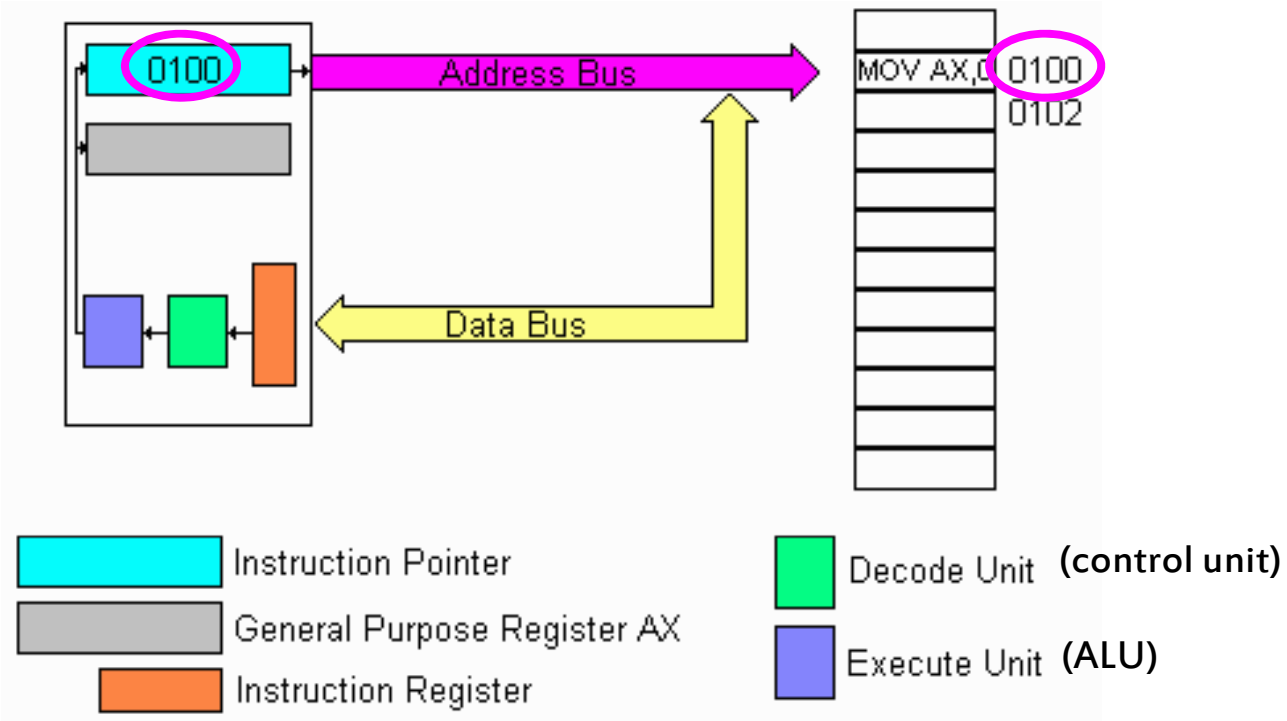
컴퓨터의 수행 단계

1. The control unit fetches the next coded instruction from memory.
2. The instruction is decoded into control signals.
3. The control signal tell the appropriate unit (ALU, memory, or I/O devices) to perform the instruction.
4. The sequence is repeated from step 1.



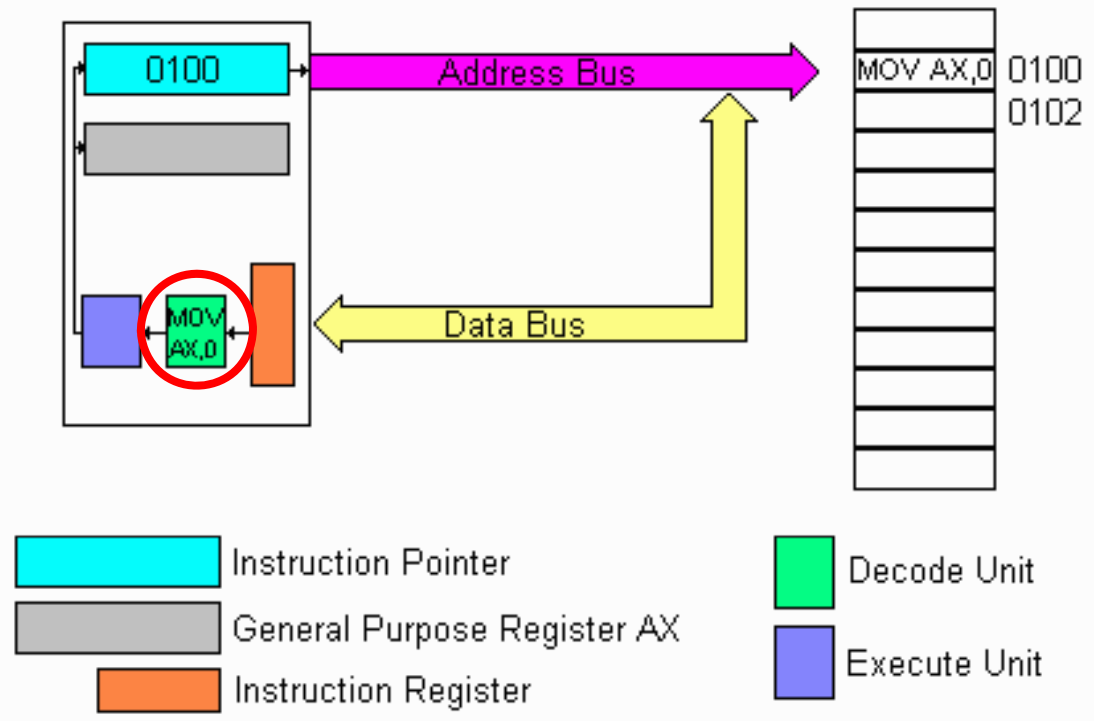
Step 1: Instruction Fetch

1. The control unit fetches the next coded instruction from memory.



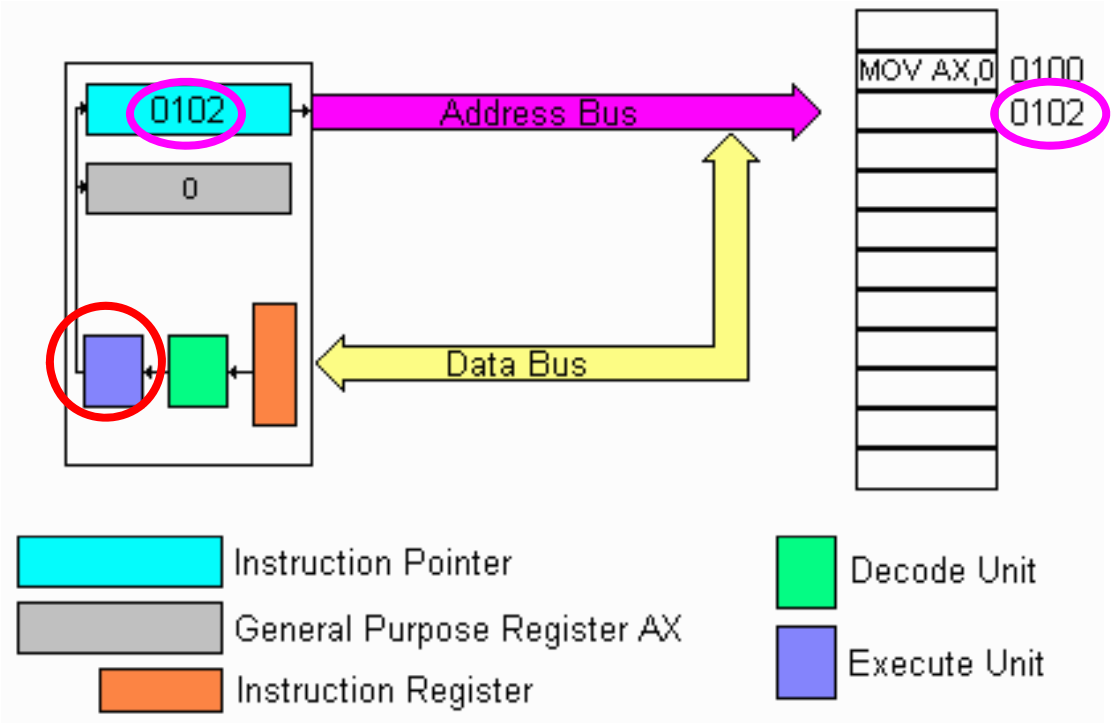
Step 2: Decode

2. The instruction is decoded into control signals.



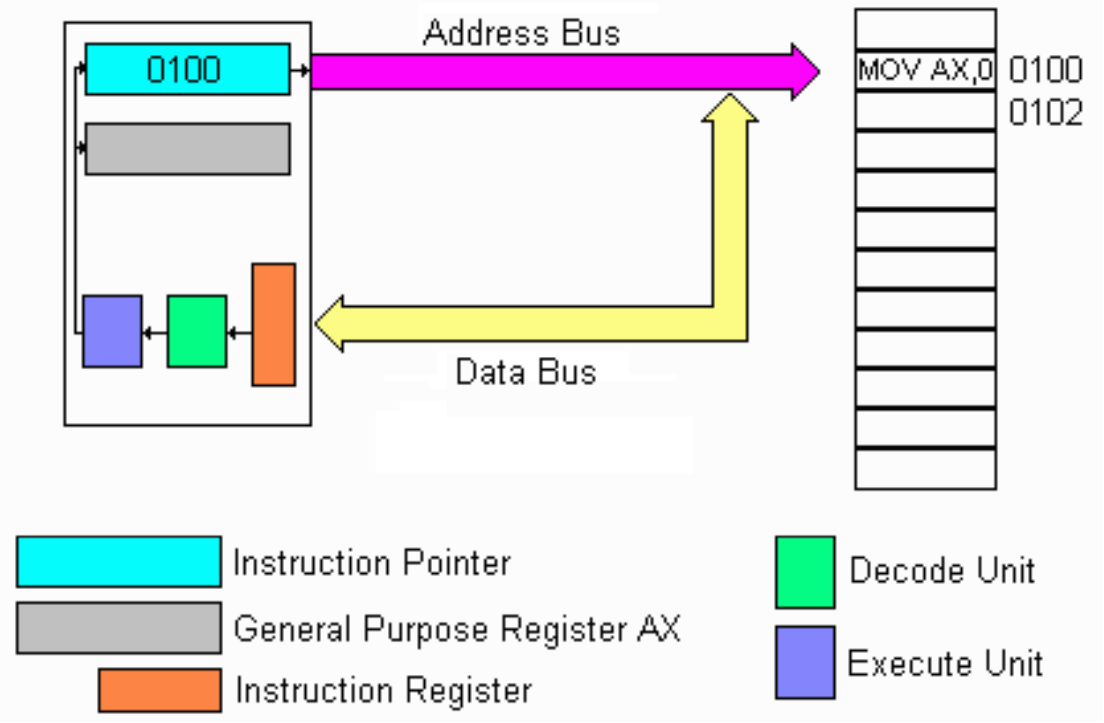
Step 3: Execute

3. The control signal tell the appropriate unit (ALU, memory, or I/O devices) to perform the instruction.



And Repeat

4. The sequence is repeated from step 1.



프로그래밍 예제

- 소팅 알고리즘
  - 버블 소트
  - 퀵 소트

- Java Tutorial Site

  http://java.sun.com/Series/Tutorial/java/threads/index.html

소팅

- 문제: 주어진 숫자를 소팅(정렬)하라.

1. 분석:

   a라는 배열에 저장된 n개의 정수들을 숫자가 작은 것부터 큰 것의 순서로 새롭게 배열하라.

2. 범용 방법론:

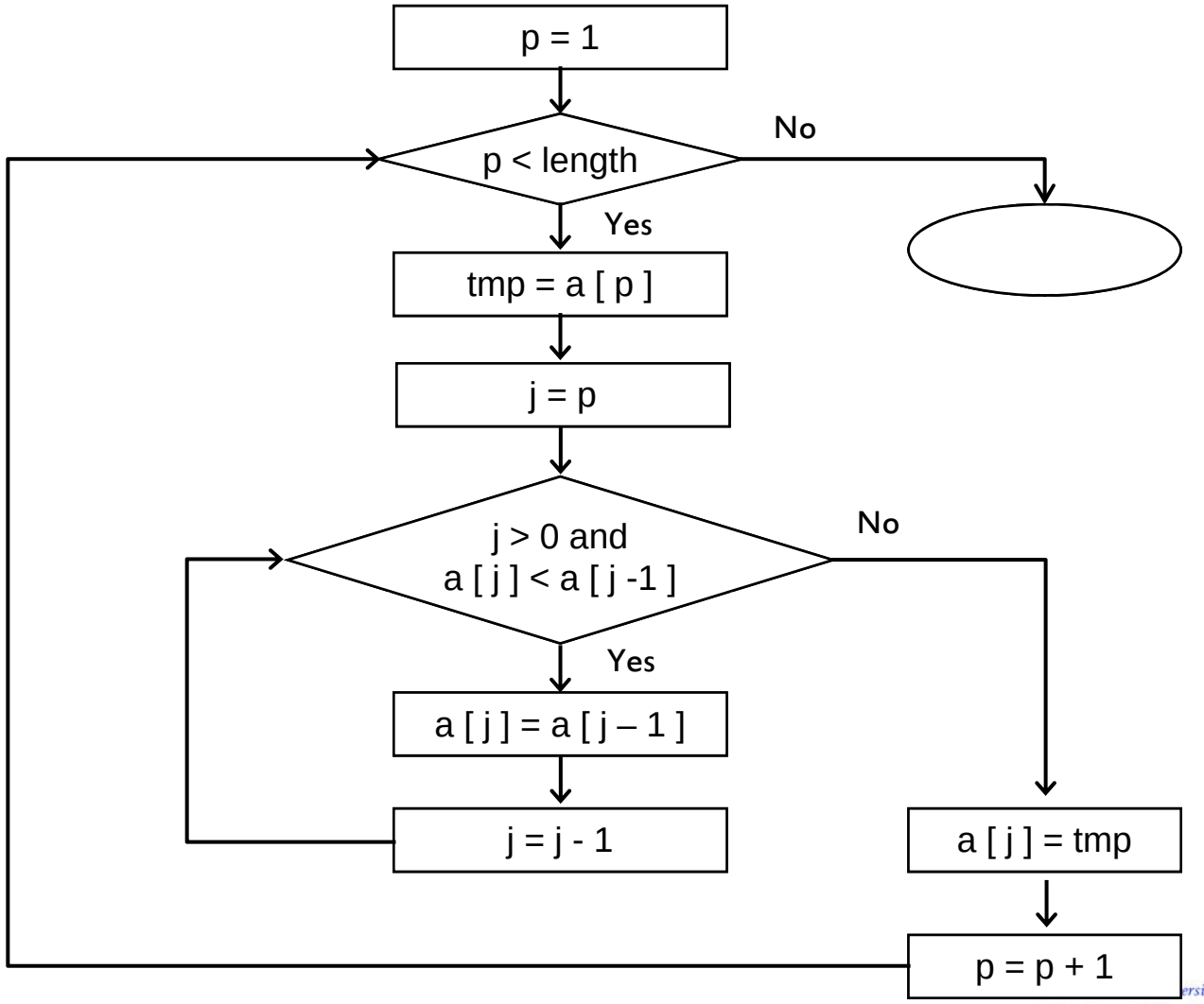
   Insertion sort, bubble sort, quick sort, ...

Basic Action of Insertion Sort

Array position	0	1	2	3	4	5
initial State:	8	5	9	2	6	3
After a[0..1] is sorted:	5	8	9	2	6	3
After a[0..2] is sorted:	5	8	9	2	6	3
After a[0..3] is sorted:	2	5	8	9	6	3
After a[0..4] is sorted:	2	5	6	8	9	3
After a[0..5] is sorted:	2	3	5	6	8	9

(shaded part is sorted)

Flow Chart of Insertion Sort



C Code of Insertion Sort

```
 1 void insertionSort( int *a, int length )
 2 {
 3     int tmp, j, p;
 4
 5      for( p=1; p<length; p++)
 6      {
 7           tmp = a[p];
 8           for( j=p; j>0 && a[j] < a[j-1]; j--)
 9                a[j] = a[j-1];
10           a[j] = tmp;
11      }
12 }
```