

C++ STUDY

Week 2
C++ & OOP

01

하나,

C++ 에 대해



02

두울,

기본 입출력



03

세엣,

문자열

01

하나,

C++ 에 대해



02

두울,

기본 입출력



03

세엣,

문자열

C++ = C + OOP (객체지향)

절차 지향 vs 객체 지향

객체와 캡슐화 (Object & Encapsulation)

상속성 (Inheritance)

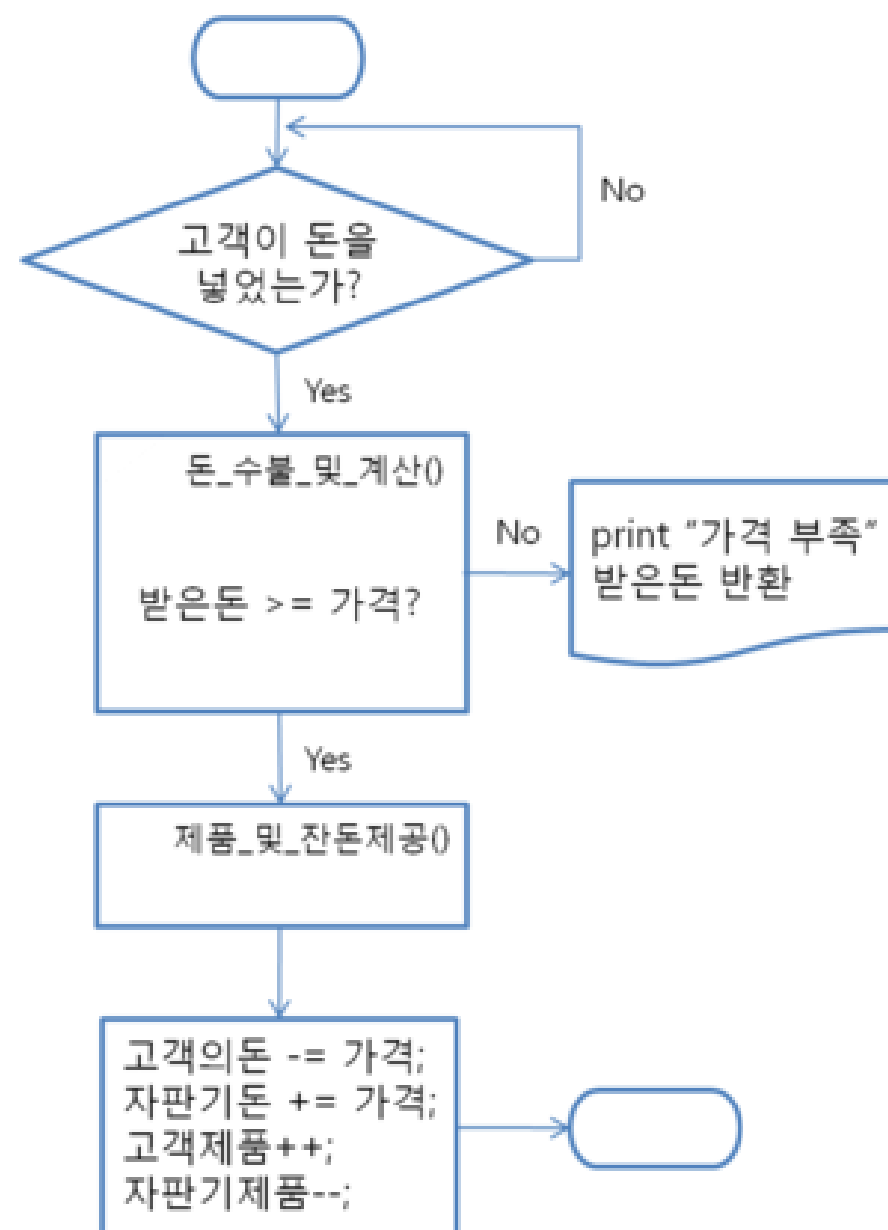
다형성 (Polymorphism)

절차 지향 vs 객체 지향

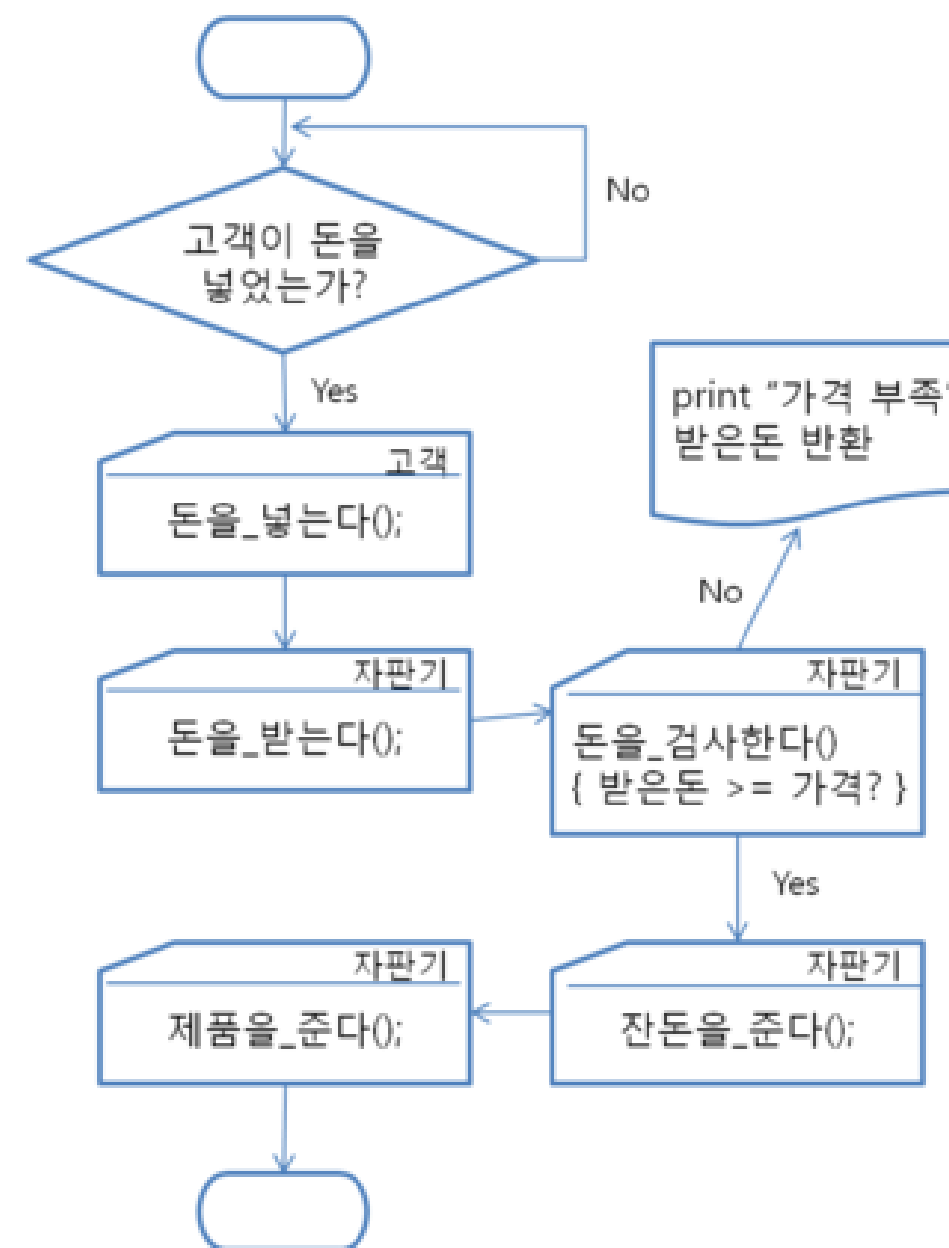
Procedural Programming

Object-Oriented Programming

절차지향 방식



객체지향 방식



절차 지향 vs 객체 지향

Procedural Programming

Object-Oriented Programming

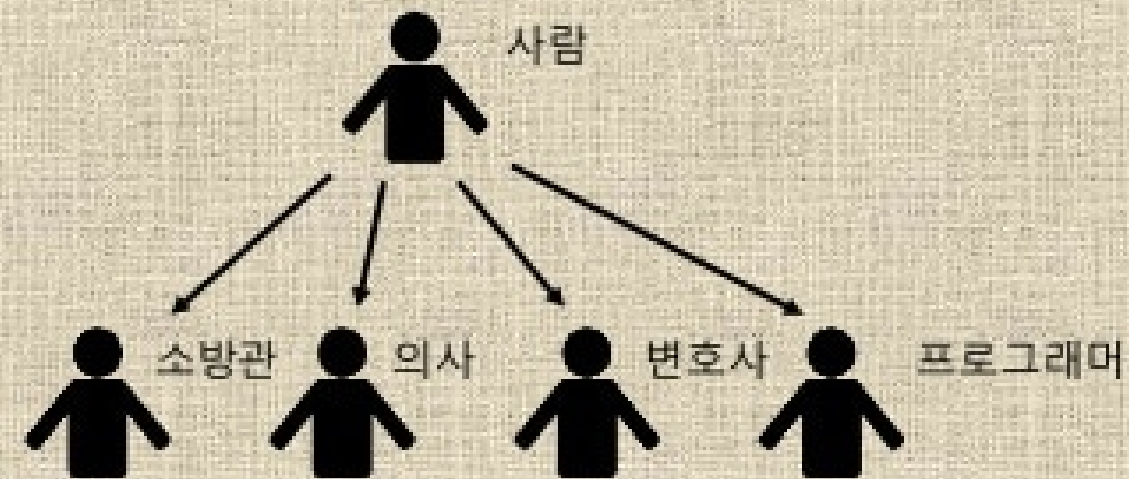
객체지향 프로그래밍 기법

- 절차지향 VS 객체지향

절차지향



객체지향



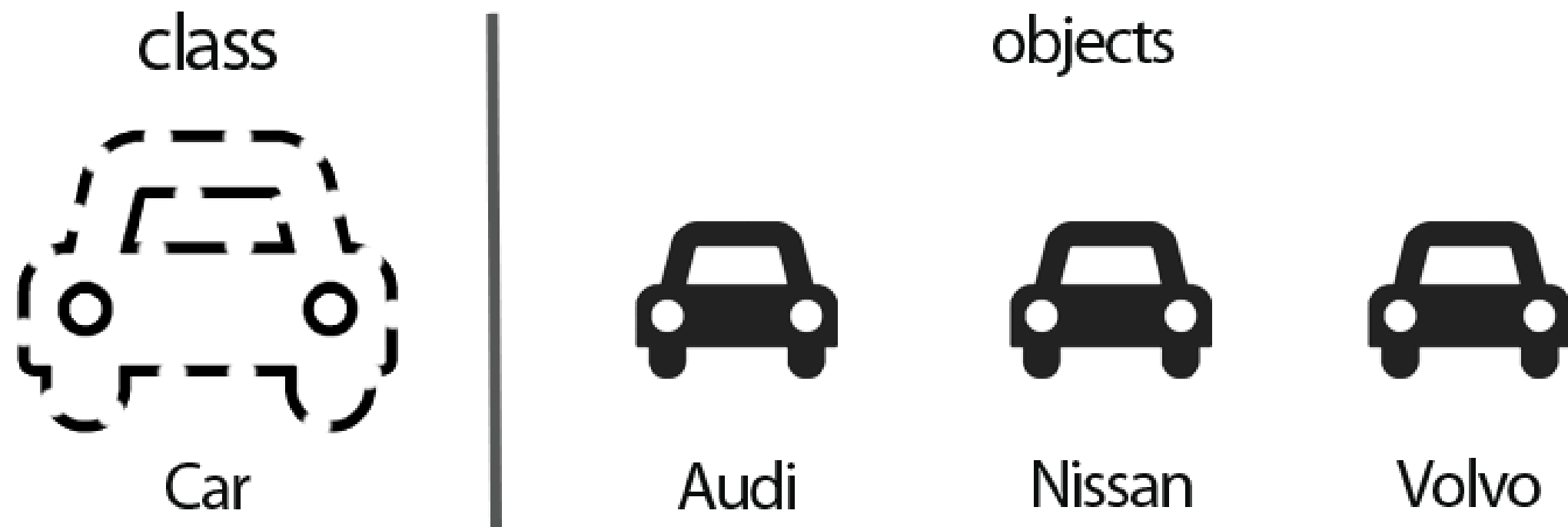
절차로 구현하게 되면 사람이라는 정의를 개개의 객체에 맞춰 모두 만들어줘야 한다.
하지만 이와는 달리 객체지향의 경우 사람이라는 것을 통해 하위를 구현하면 되므로 훨씬 코드가 부드러워진다.

By Kim Jong Uk

객체와 캡슐화 (Object & Encapsulation)

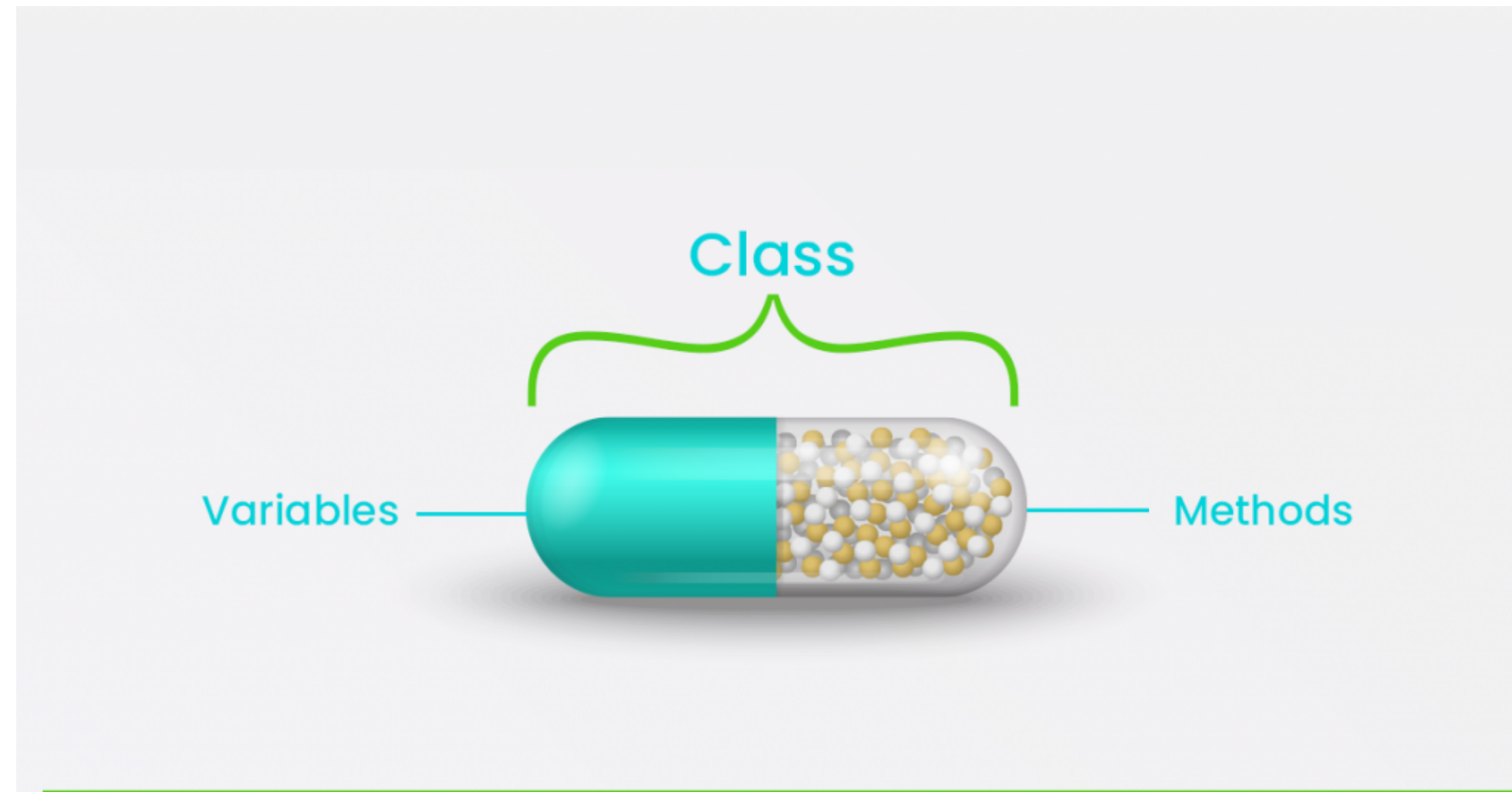
클래스 (class) : 객체를 정의하는 틀

객체 (object) : 클래스라는 틀에서 생겨난 '실체' (instance)



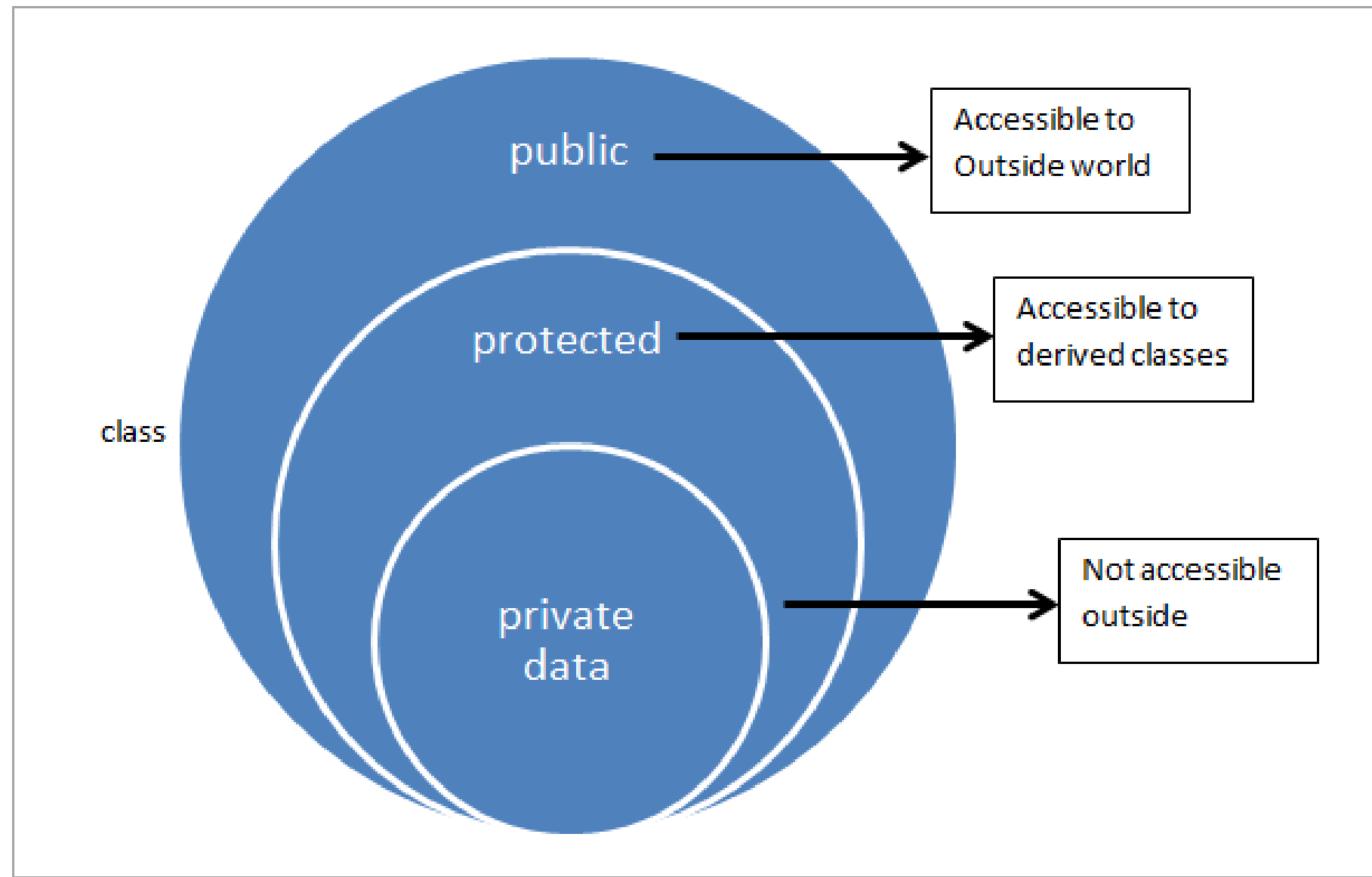
객체와 캡슐화 (Object & Encapsulation)

캡슐화 (Encapsulation) :
데이터를 캡슐로 싸서 외부의 접근으로부터 데이터 보호



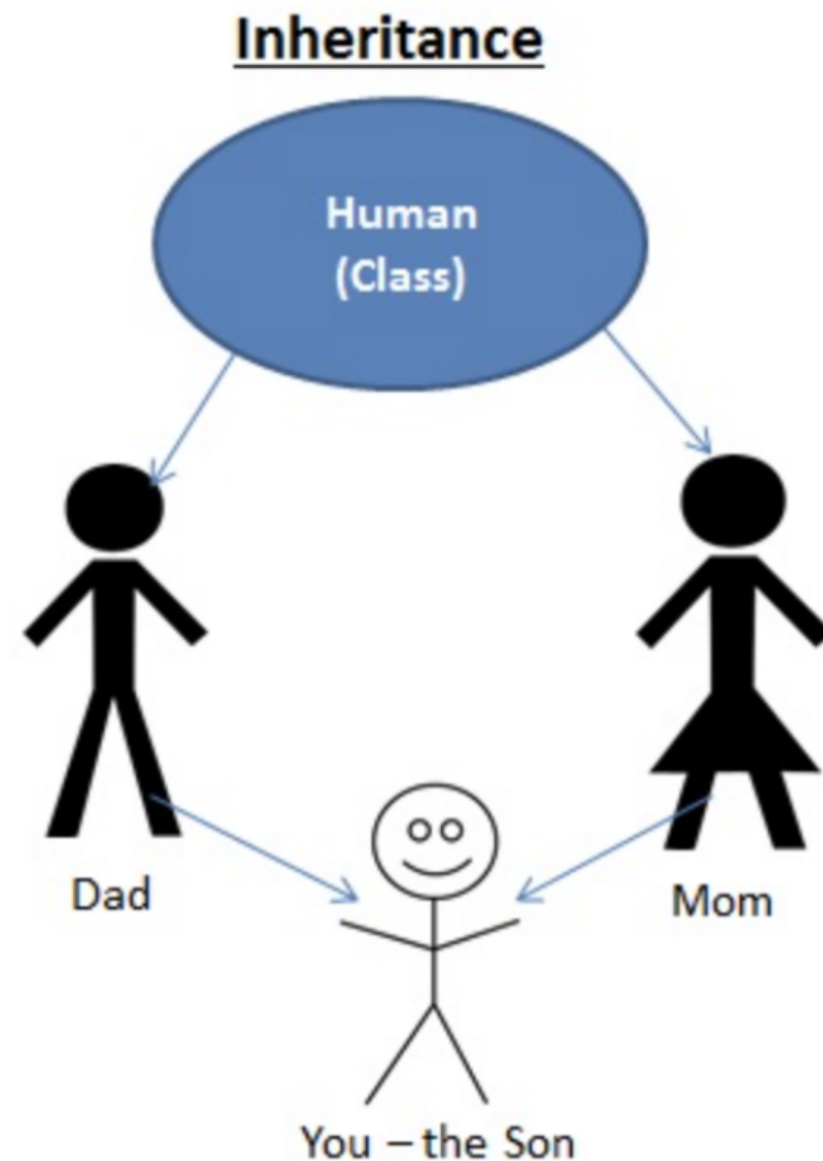
객체와 캡슐화 (Object & Encapsulation)

정보 은닉 (information hiding)



상속성 (Inheritance)

사전적 의미 : 일정한 친족적 관계가 있는 사람 사이에 한 쪽이 사망하거나 법률상의 원인이 발생하였을 때 재산적 또는 친족적 권리와 의무를 계승하는 제도



상속성 (Inheritance)



전화 걸기()
전화 받기()

+



인터넷 접속()
게임하기()

+

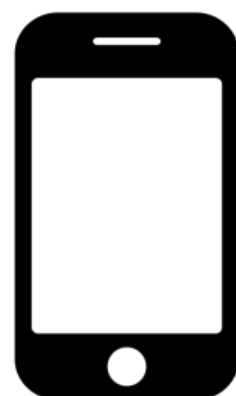
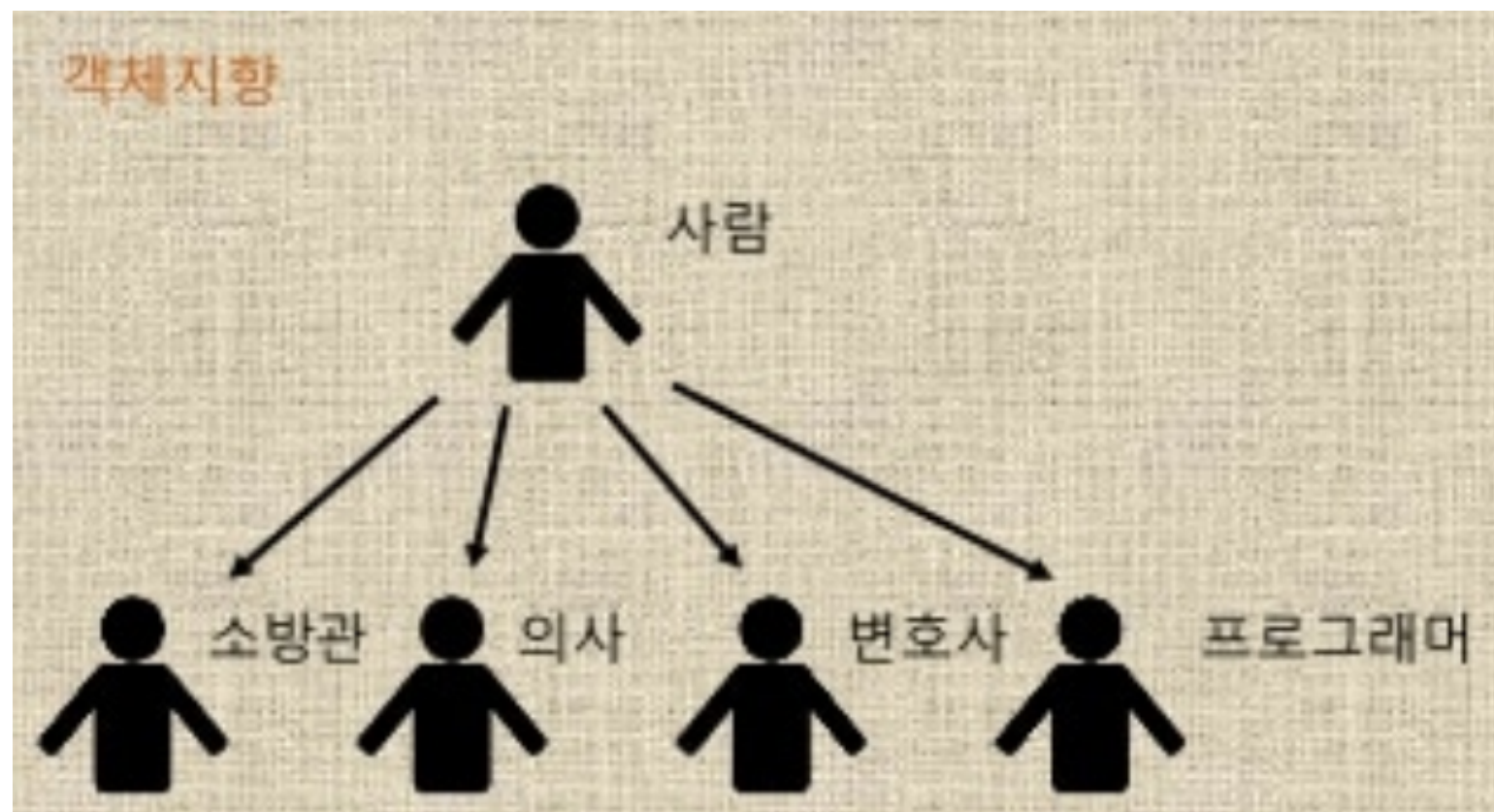


사진 찍기()
어플 설치()

다형성 (Polymorphism)

하나의 기능이 경우에 따라 서로 다르게 보이거나 다르게 작동하는 현상



다형성 (Polymorphism)

오버로딩 : 동일한 이름이지만 다른 기능을 갖거나 다른 매개변수를 갖는다.

오버라이딩 : 하위 클래스에서 상위 클래스의 메소드를 재정의 한다.

01

하나,

C++ 에 대해



02

두울,

기본 입출력



03

세엣,

문자열

기본 입출력

```
#include <iostream> // input output stream <- cout과 << 연산자 포함

int main()
{
    std::cout << "Hello C++\n"; // Hello C++를 출력하고 다음 줄로 넘어감
    std::cout << "첫 C++ 예제입니다." << std::endl;

    return 0; // 0을 반환하고, 프로그램 종료
}
```

기본 입출력

```
#include <iostream> // input output stream <- cout과 << 연산자 포함
```

<iostream> 헤더 파일을 읽어 현재 프로젝트에 삽입.

iostream -> C++ 표준 입출력을 위한 클래스와 객체가 선언되어 있다.

기본 입출력

```
std::cout << "첫 C++ 예제입니다." << std::endl;
```

cout : 스크린 장치와 연결된 C++ 표준 출력 스트림 객체

<< : 스트림 삽입 연산자, 오른쪽 피연산자 데이터를 왼쪽 스트림 객체에 삽입

endl : **\n**을 스트림 버퍼에 넣고 스트림 버퍼 안의 모든 데이터 출력

std::cout << "첫" << " C++" << "예제입니다." << std::endl; 도 가능하다

가능한 이유? Chaining

실습(1)

**원의 면적을 구하는 함수를 만들고
main 함수에서 반지름이 5인 원의 면적 구하는 프로그램.**

기본 입출력

namespace (이름공간) : 같은 이름을 구분해준다.

namespace (이름공간) :: identifier(이름)

A.cpp

```
void run()
{
    ...
}
```

namespace A

```
{
    run()
}
```

B.cpp

```
void run()
{
    ...
}
```

namespace B

```
{
    run()
}
```

main.cpp

```
A::run();
B::run();
```

기본 입출력

```
std::
```

C++ 표준 라이브러리에서 선언된 이름(identifier)을 사용할 때 사용

```
using namespace std;
```

using 지시어 : 이름 공간 접구어를 생략할 수 있게 해준다.

실습(2)

아까 만든 프로그램에서 using 지시어를 사용하여 재작성

기본 입출력

```
#include <iostream>
using namespace std;

int main()
{
    int num = 0;

    cout << "숫자 입력 : ";

    cin >> num;

    cout << "입력된 숫자 : " << num << endl; // %d 사용할 필요 없이 변수를 쓰면 된다.
}
```

기본 입출력

```
cin >> num;
```

cin : C++ 표준 입력 스트림 객체

**>> : 스트림 추출 연산자, 왼쪽 피연산자인 스트림 객체로부터 데이터를 읽어
오른쪽 피연산자에 지정된 변수에 삽입**

실습(3)

반지름을 입력받고 해당 반지름에 대한 원의 면적을 출력하는 프로그램

01

하나,

C++ 에 대해

02

두울,

Object-Oriented Programming

03

세엣,

문자열

문자열

C-string : C 언어에서 문자열을 표현하는 방법

string class : 문자열을 string 객체로 다루는 방법

문자열 C-string 을 이용한 방법

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char name[10] = { 'T', 'e', 'd', 'd', 'y' }; // 10크기의 배열이므로, 9개의 영문자만 저장 가능하다.

    cout << "제 이름은 " << name << " 입니다." << endl;
}
```

strlen(), strcmp(), strcpy() 등등 전부 사용가능하다!

실습(4)

C-string을 이용하여 암호가 입력되면 프로그램이 종료되는 프로그램 작성

암호는 "C++"로 설정, 최대 11의 크기의 배열 선언, strcmp() 함수 활용

**암호가 틀렸다면 틀렸다는 메시지 출력 후 재입력,
암호가 맞다면 암호가 맞다는 메시지 출력 후 프로그램 종료**

문자열 C-string 을 이용한 방법

cin>> 으로 입력을 받을 때의 허점 : 빈칸을 만나면 문자열 입력을 종료한다.

cin.getline() 을 이용

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char name[10] = { 'T', 'e', 'd', 'd', 'y' }; // 10크기의 배열이므로, 9개의 영문자만 저장 가능하다.
    cout << "제 이름은 " << name << " 입니다." << endl;

    char email[100] = {};
    cin >> email;
    cout << "이메일 : " << email << endl;

    cin.ignore(100, '\n'); // 입력 버퍼 비워주기

    cin.getline(email, 100, '\n');
    cout << "이메일 : " << email << endl;
}
```

문자열 string class를 이용한 방법

string 의 특징 : 문자열의 크기에 제약이 없다. 다루기 쉽다.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string song("Love Yourself");
    string bieber("Justin Bieber");
    string singer;

    cout << "가수 입력 : ";
    getline(cin, singer);

    cout << song << "을 부른 가수는 " << singer;
}
```

실습(5)

**string을 이용하여 Love Yourself를 부른 가수를 입력 받고,
가수가 맞는지 알려주는 프로그램**

두 string 비교는 == 연산자로 가능

맞으면 맞다고 출력, 틀렸다면 정답 출력

과제

1. 가위바위보 프로그램 만들기

두 사람이서 가위바위보를 한다. 두 사람의 이름은 영수, 철수라고 한다. 먼저 "영수>> "를 출력하고, "가위" "바위" "보" 중 문자열을 입력받고, "철수>> "를 출력하고, "가위" "바위" "보" 중 문자열을 입력받고, 누가 이겼는지 판단하여 문자열을 출력한다. 비길 경우 비겼습니다를 출력.

Q&A 및 잡담(?)

감사합니다!