

# 프로그래밍의 사고 (소프트웨어 개발의 사고) Computational Thinking

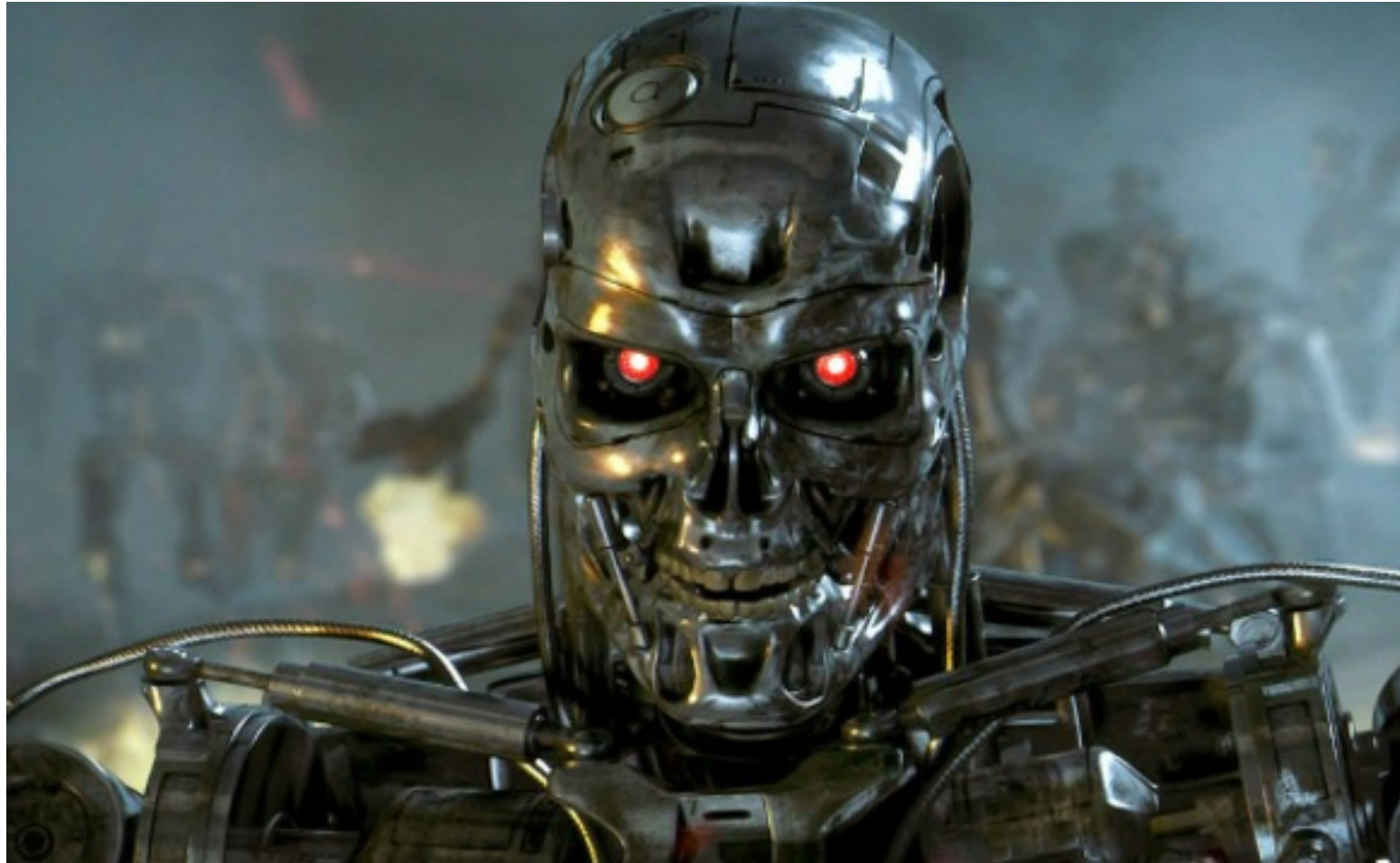
Updated in 2020.04

한대희

# Computational Thinking

# 심부름 robot

- 새우깡을 사오라고 시켰다.
- 로봇트는 무사히 새우깡을 사올 수 있을까?



# 담배 사러 가기 단계

- 문을 열고 나간다
- 엘리베이터를 타고, 버튼을 누르고, 내려서 나간다.
- 편의점을 찾아간다
- 편의점 내부에서 새우깡을 찾는다
- 돈 1만원을 내고 거스름돈을 받아 온다.

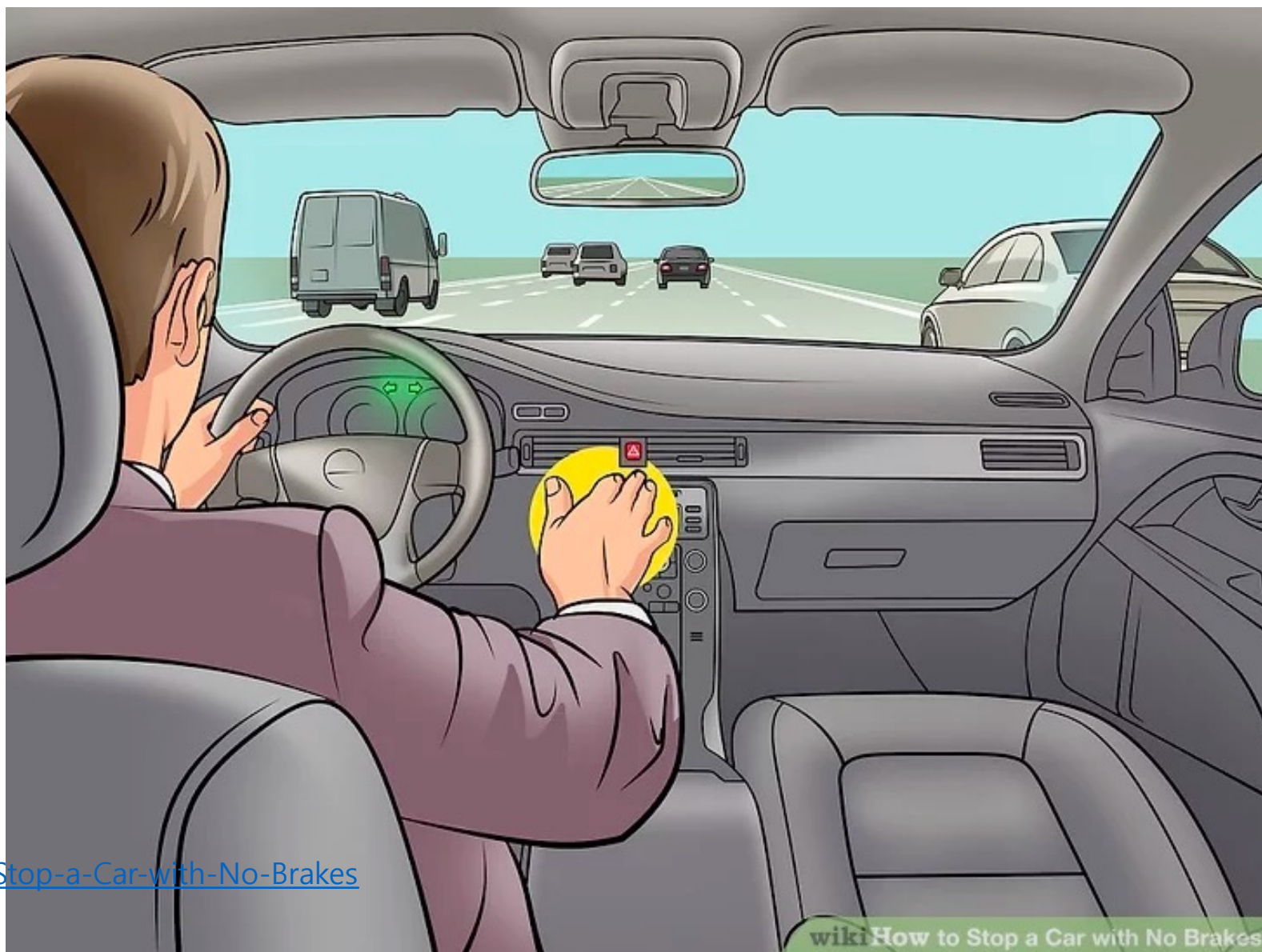
# Divide & Conquer

- 전체 일을 작은 단위로 나누고, 각각의 완성 방법을 생각하라
- 일을 쪼개기 : Module 단위로 나누기. 모듈화.
- 업무 분담과 유사함
- Role/Responsibility를 나누기

아래 2개 중 자동차는?







<https://www.wikihow.com/Stop-a-Car-with-No-Brakes>





# 의자란 무엇인가? - 다음 중 의자가 아닌 것은?



# Abstraction (추상화)

- 추상화는 복잡한 내부 기능보다는 핵심적 기능에 집중.
- 실제적 객체(사물)들에서 공통적이고 핵심적인 기능 및 속성값을 찾아내어 개념화 하는 작업.
- 일반화(generalization) 와 유사하다고 함.
- [https://en.wikipedia.org/wiki/Abstraction\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Abstraction_(computer_science))

# Abstraction

- 엘리베이터 버튼은
- 엘리베이터의 여러 기능 중 일반 사용자에게
- 필요한 기능을 추상화하여 보여준다.



# Abstraction

- 자동차를 추상화한다면
- Brake, Accelerator, Steering wheel 등이 필요
- Methods
  - brake(on), brake(off)
  - accel(1), accel(3), accel(0)
  - turn(-30), turn(+30)
  - turn\_left(30), turn\_right(30)
  - Brake와 Accel을 동시에 밟을 수 없도록 구현해야함 → 예) exception 발생하도록 구현

Real World

A Car



Abstraction



Class CAR

Properties (Data Members):

Date when built

Capacity (PS)

Serial number

...

Methods (Member functions):

to run, to brake,

to park, to turn off

...

Instantiation



Objects

**car1**

Properties:

Date when built = 1990

Capacity = 100

Chassis number = 11111

...

Methods

...

**car2**

Properties:

Date when built = 2000

Capacity = 200

Chassis number = 22222

...

Methods

...

...

# Class/Object

- 사람의 attributes: name, age, weight, ...
- Car의 attributes: name, brand, speed, x, y, ...

## Introduction to object-oriented design

Class name
Attribute 1 Attribute 2 ...
Method 1 Method 2 ...

Car
Make Model Doors BodyLength EngineSize Color Speed
Accelerate() Stop() Brake() Turn(direction)



# Class/Object 관련 내부 메모리 할당

- Class를 정의하면
- Method (code)를 저장하기 위한 메모리는 클래스별로 할당
- Object를 저장하기 위한 메모리는 객체별로 할당
- Object에 대한 메모리는 객체의 속성값(멤버 변수들)을 저장

Cars	
<i>data:</i>	
1.	name
2.	model
3.	company
4.	date_of_manufacture
5.	engine
<i>functions:</i>	
1.	turn_engine()
2.	brake()
3.	accelerate()
4.	clutch()
5.	change_gears()
6.	blow_horn()

# 객체(object)의 3요소

- 상태 (State) – 객체내의 attribute(속성)값(들). Member variables.
- 행위 (Behavior) – 상태값을 변화시킬 수 있는 수단, Method, Function
- 식별 (Identity) – 파이썬의 id(), 개체 식별자

**Name**  
(Identifier)

**Variables**  
(Static attributes)

**Methods**  
(Dynamic behaviors)

Student	Circle	SoccerPlayer	Car
name gpa	radius color	name number xLocation yLocation	plateNumber xLocation yLocation speed
getName() setGpa()	getRadius() getArea()	run() jump() kickBall()	move() park() accelerate()

**Examples of classes**

# Library / Package

- Windows / Unix 시절에는 Library가 일반적인 용어
- Python, R 등에서는 모듈들을 package 라고 부름
  - 3<sup>rd</sup> party가 활발히 개발하여 제공함

# API

- Library/Package가 제공하는 클래스 및 메소드의 규격
  - 파라미터, 리턴값 정의
- Restful API
  - HTTP 기반의 API.
  - 브라우저 및 앱이 통신을 통해 서버에 데이터를 요청 및 수정하는 기능

# 프로그래밍 용어



# IF (조건문)

# Loop (반복문)

- For 루프 → 반복 횟수
- While 루프 → 반복 조건

# Exception

- 예상하지 못한 상황이 발생했을 때 처리 방안
- Exception Handling 이라고 함

# Function/Method

# Parameter/Argument

Return값

C



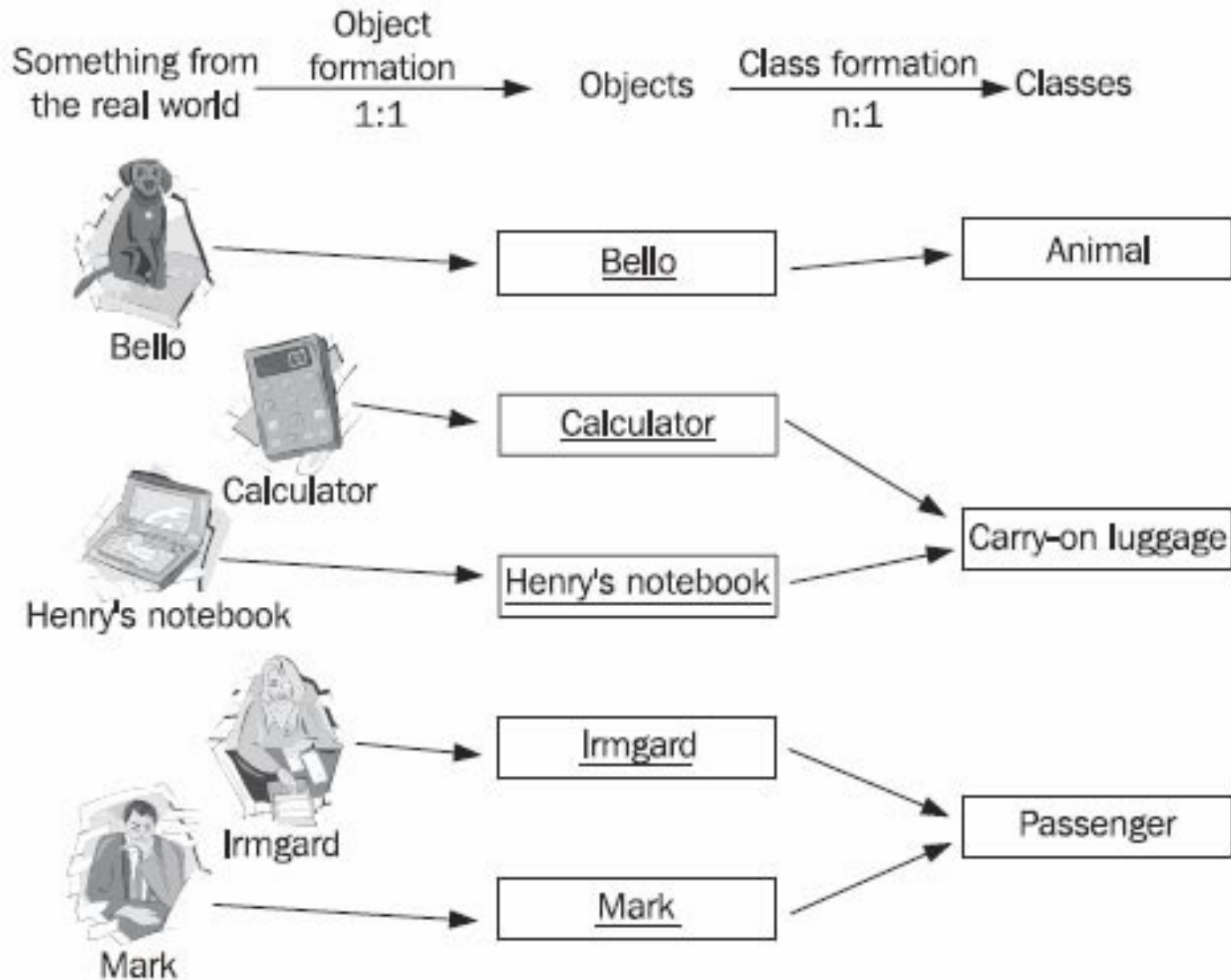
# Variable, 변수

c

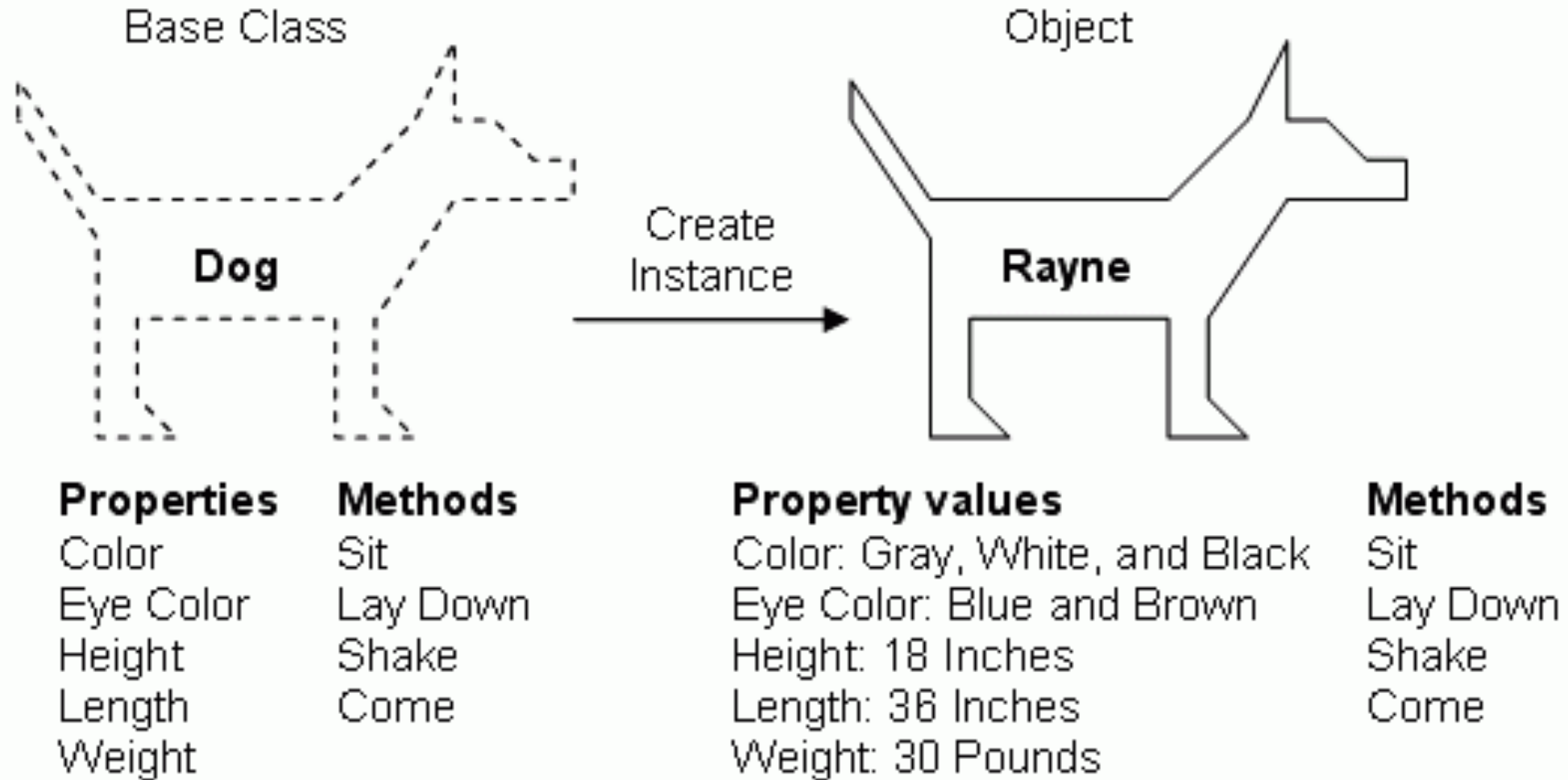
# Data Type

- 데이터 타입에 따라 size, operation 달라진다
- 숫자
  - 정수
  - 실수
- 문자열
- Array
- 구조체(structure)
- Hash : key-value 조합
  - 다른 이름으로
  - Map
  - Dictionary

# Class/Object



# Class/Object/Property/Method



**The End**