



# OOAD

## บทที่ 2 : แนะนำ Object Orientation เบื้องต้นและ Abstractions



Asst. Prof. Dr. Nattapong Songneam  
<http://www.siam2dev.com>  
[Dr.nattapong\\_s@hotmail.com](mailto:Dr.nattapong_s@hotmail.com)  
[siam2dev@hotmail.com](mailto:siam2dev@hotmail.com)  
[xnattapong@hotmail.com](mailto:xnattapong@hotmail.com)



# รายละเอียดของโครงการ

ลำดับ	รายการ	คะแนน	กำหนดส่ง
1	แบ่งกลุ่ม ห้องละ 2-3 กลุ่ม ตามจำนวน นศ.	-	สัปดาห์ที่ 2
2	กำหนดชื่อเรื่องโครงการ	-	สัปดาห์ที่ 2
3	ทำการวิเคราะห์ เพื่อหา Business Rule ของระบบงาน	10	สัปดาห์ที่ 4
4	1) กำหนดความต้องการ 2) ทำเอกสาร SRS 3) ทำคสิปวิดีโอ	10	สัปดาห์ที่ 9
5	1) การวิเคราะห์และออกแบบระบบเชิงวัตถุ 2) จัดทำเป็นรูปเล่มรายงานฉบับสมบูรณ์	10	สัปดาห์ที่ 15



# สยามทูเดฟ

<http://www.siam2dev.net>

โดย อาจารย์ ดร. นัฐพงศ์ ส่งเนียม

การวิเคราะห์และออกแบบระบบเชิงวัตถุ

# Abstractions

communication diagram

inheritance

OOSE DOMAIN

polymorphism

CASE TOOLS

class diagram

# UML

# OOAD

use case diagram

activity diagram

encapsulation

compnent diagram

classification abstraction

sequence diagram

generalization abstraction

class&object OO deployment diagram agile

object oriented analysis and design unified process

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

## บทที่ 2 : แนะนำ Object Orientation และ Abstraction

- วัตถุประสงค์
  - เพื่อให้เข้าใจเกี่ยวกับแนวคิด Object Orientation
  - เพื่อให้เข้าใจหลักการการพัฒนาระบบงานด้วย Object-Oriented Software Engineering
  - เพื่อให้รู้จักกับคำศัพท์พื้นฐาน ได้แก่ Problem Domain , Object , Class , Behavior , Function , Abstractions, Real world, Computer World, OOA , OOD , OOP เป็นต้น



# Agenda

## แนะนำ Object Orientation เบื้องต้นและ Abstractions

### Reviews

#### 2.1 OO คืออะไร

- ความสัมพันธ์ (Relationship)
- ปฏิสัมพันธ์ (Interaction)

#### 2.2 Problem Domain

- Object และ Classes
- class หรือ Abstract Object
- Abstraction & Instantiation
- Attributes และ Functions
  - Attribute
  - Functions

#### 2.3 OOSE

- OOA (Object Oriented Analysis)
- OOD (Object Oriented Design)
- OOP (Object Oriented Project)

#### 2.4 Abstractions



# การพัฒนาาระบบ/ ซอฟต์แวร์

## 1. แบบดั้งเดิม Traditional / แบบโครงสร้าง Structural Programming

เริ่มประมาณ ปี 1960 -1990

### – Algorithm

- ขั้นตอนวิธี เน้น การแก้ปัญหาโดยแบ่งออกเป็นขั้นตอน ต้องคิดหรือหาวิธีใหม่

### – ภาษาเชิงโครงสร้าง Structural Programming (c, pascal, basic)

## 2. เชิงวัตถุ OO: Object Oriented

- ภาษาเชิงวัตถุ เช่น c++ , java , vb , vc# , Python , PHP  
...

เริ่มประมาณ ปี 1980 เป็นต้นมา



# Algorithm

ขั้นตอนวิธี เน้น การแก้ปัญหาโดยแบ่งออกเป็นขั้นตอน ต้องคิดหรือหาวิธีใหม่

- วิชา การเขียนโปรแกรมและขั้นตอนวิธี / วิชาการเขียนโปรแกรม  
ภาษาคอมพิวเตอร์ 1

– วิธีประมวลผล

**จงหาอายุเฉลี่ยของ นศ. ทั้งห้อง**

Algorithm : .....Calculate\_Average\_Age

1. START / เริ่มต้น
2. Initialization Stage ขั้นตอนการเริ่มต้น การกำหนดค่าเริ่มต้นของตัวแปร การนิยามตัวแปรต่าง ๆ การเคลียร์ค่า เตรียมความพร้อม
3. INPUT การรับข้อมูล การอ่านค่า การอ่านไฟล์เข้ามา
4. PROCESS การคำนวณ การทำงานต่าง ๆ
5. OUTPUT การแสดงผลลัพธ์ของการทำงาน
6. STOP

# Algorithm

## จงเขียนอัลกอริทึมของการหาพื้นที่วงกลม

Algorithm : การคำนวณหาพื้นที่วงกลม / Circle Area Calculation

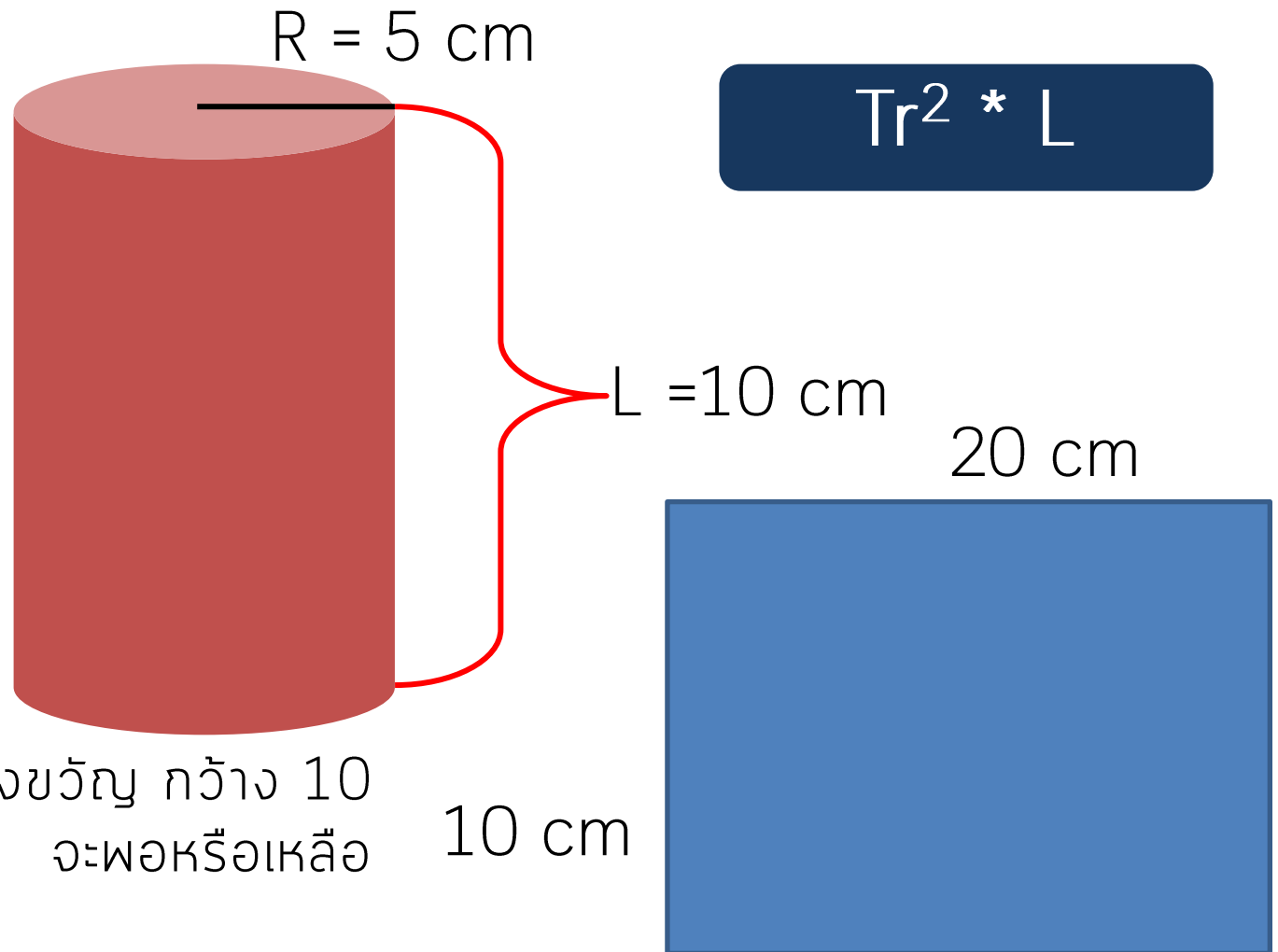
- 1. START
2. Initialization Stage ขั้นตอนการเริ่มต้น การกำหนดค่าเริ่มต้นของตัวแปร การนิยามตัวแปรต่าง ๆ การเคลียร์ค่า เตรียมความพร้อม
3. INPUT การรับข้อมูล การอ่านค่า การอ่านไฟล์เข้ามา
- 4. PROCESS การคำนวณ การทำงานต่าง ๆ
5. OUTPUT การแสดงผลลัพธ์ของการทำงาน
- 6. STOP

# จงวิเคราะห์อินพุตของโจทย์ต่อไปนี้

- จำนวนพื้นที่วงกลม
  - $PI * R * R$
  - ค่ารัศมี R
- จำนวนพื้นที่สี่เหลี่ยมคางหมู
  - $\frac{1}{2} * (\text{ผลบวกของด้านคู่ขนาน}) * \text{สูง}$
  - ด้านคู่ขนานทั้งสอง และ สูง
- จงหาจำนวนเฉพาะ (Prime number) ของตัวเลขตั้งแต่ 1..100
  - ไม่มีอินพุต
- จงหาระยะทาง 1 ปีแสง (Light Year)
  - ค่าคงตัว หรือ ค่าคงที่ไม่ต้องรับ
  - เรารู้ว่า อัตราเร็วของแสง เท่ากับ  $3 * 10^8 \text{ m/s}$
- จงหาเงินเดือนมากที่สุดของพนักงาน 10 คน
  - \*\*\* ยิ่งข้อมูลนำเข้าน้อย ก็จะทำให้ผู้ใช้งานโปรแกรมง่ายขึ้น**

# จงคำนวณปริมาตร ทรงกระบอกนี้

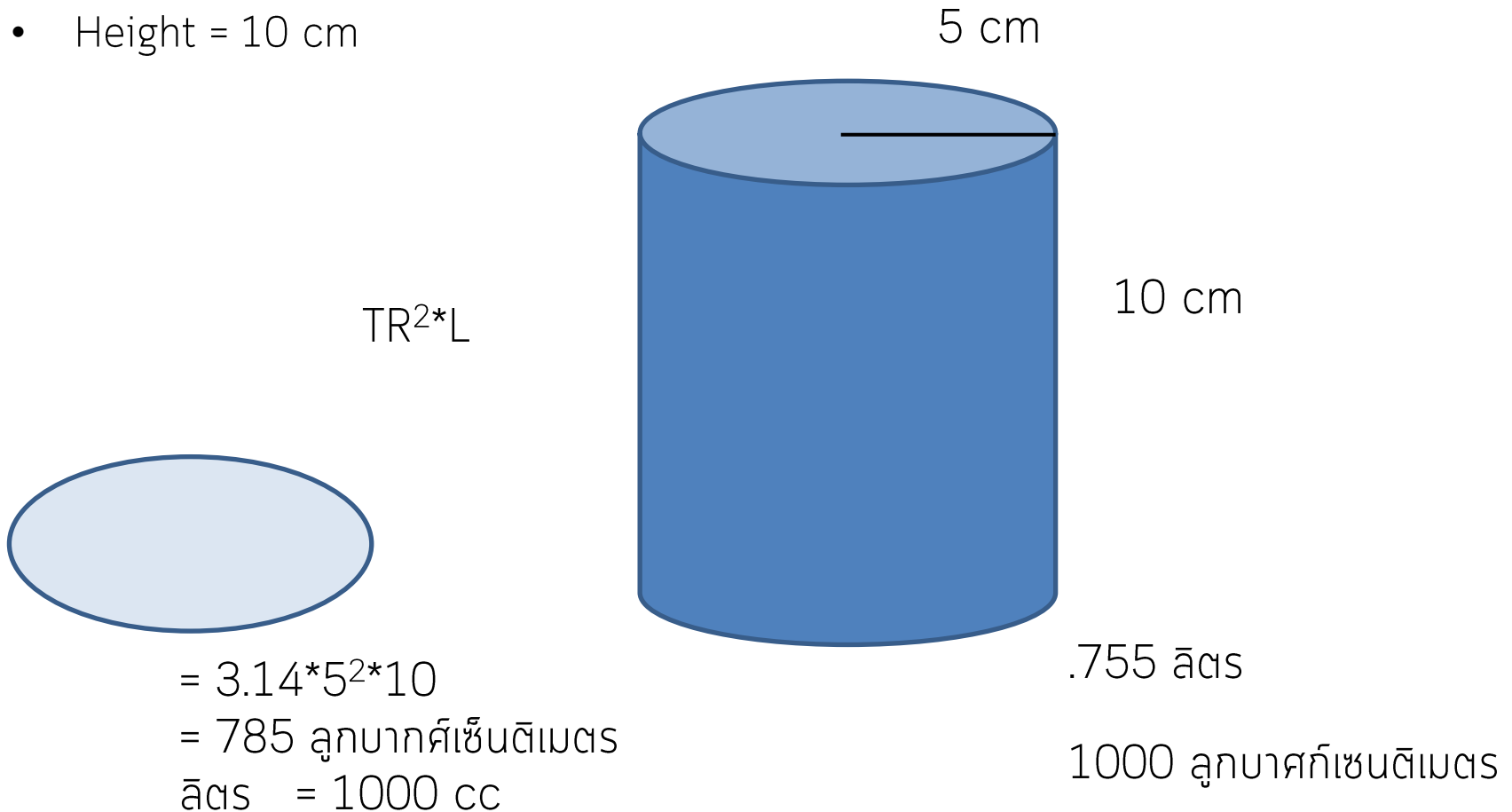
- Radius
- H หรือ L



ถ้านำกระดาษห่อของขวัญ กว้าง 10 cm ยาว 20 cm จะพอหรือเหลือ หรือขาดอยู่เท่าใด

# อยากทราบว่าทรงกระบอกนี้สามารถบรรจุน้ำได้ที่ลิตร

- Radius = 5 cm
- Height = 10 cm





บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

วิธีคิด

- พื้นที่ผิว ทรงกระบอก = ด้านบน + ด้านล่าง + ด้านข้าง
- $2 * (Tr^2) + 2Tr * L$
- $2 * (3.14 * 25) + 2(3.14*25)*10 = 1927 \text{ cm}^2$
- พื้นที่กระดาษ =  $10 * 20 = 200 \text{ cm}^2$
- $1927 - 200 = 1727$
- ขาด  $1727 \text{ cm}^2$

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

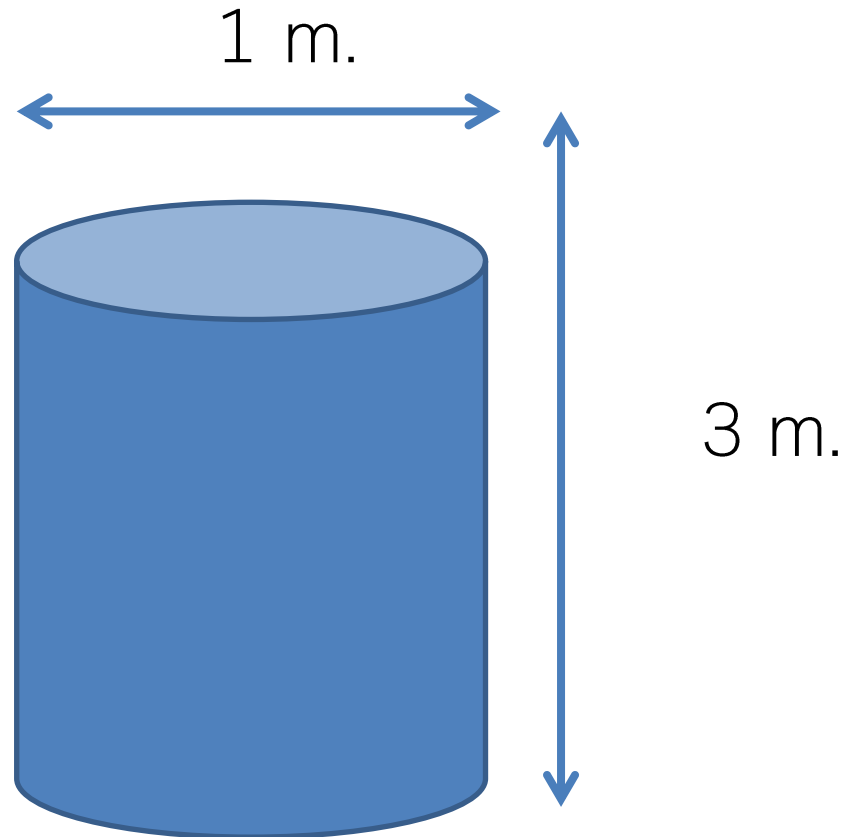
PSEUDO-CODE

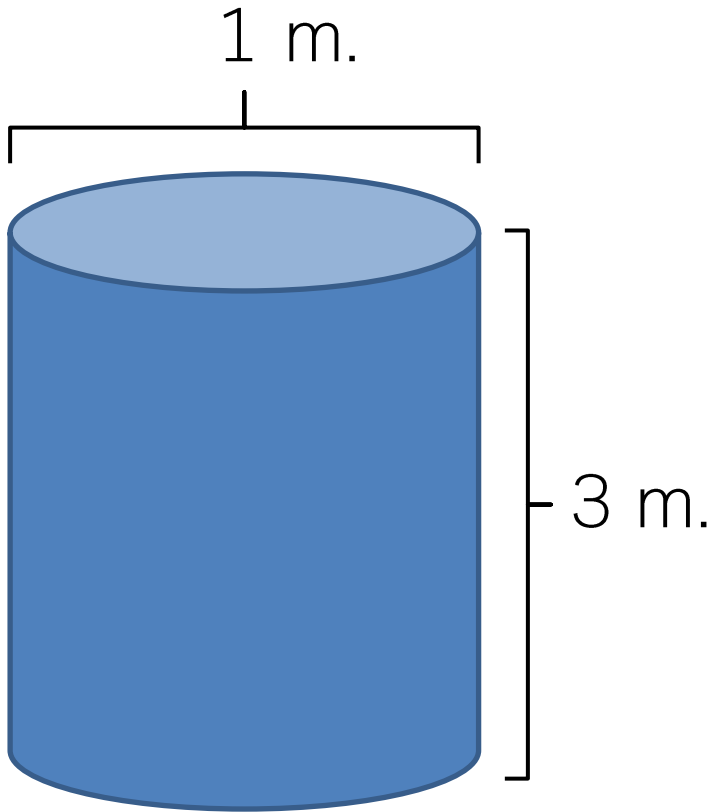
1. START
2. คำนวณ พท. ฝิว ทรงกระบอก
3. คำนวณ พท. กระดาษห่อของขวัญ
4. คำนวณ ผลต่าง ของทั้งสอง พท.
5. แสดงผล
6. STOP

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Oriented and Abstractions)

1. จงแสดงวิธีคำนวณหาว่าถังน้ำนี้  
บรรจุน้ำได้ที่ลิตร

2. จงเขียนขั้นตอนวิธี (Algorithm)  
เพื่อคำนวณหาว่าถังน้ำนี้บรรจุน้ำได้ที่  
ลิตร





วิธีคิด

1. สูตรคำนวณปริมาตรทรงกระบอก  $\pi r^2 * L$

2. คูณด้วย 1000

3. แสดงผล

1. จงแสดงวิธีคำนวณหาว่าถังน้ำนี้บรรจุน้ำได้ที่ลิตร

2. จงเขียนขั้นตอนวิธี (Algorithm) เพื่อคำนวณหาว่าถังน้ำนี้บรรจุน้ำได้ที่ลิตร

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

หลักการทำงานของคอมพิวเตอร์



- ค่าคงที่ constant เช่น ค่า PI = 3.14

- ค่าคงที่ constant เช่น ค่า VAT = 0.07

- จำนวน วันใน 1 ปี จำนวน ชั่วโมง ใน 1 วัน

# วิธีประมวลผล : ทำอย่างไร

1. เริ่มต้น
2. ....
3. ....
4. ....
5. ....
6. ....
7. ...
8. ....
9. สิ้นสุด/ จบ / เสร็จสิ้น

- **Initialization Stage**
- Input Stage
- Process Stage
- Output Stage

เขียนมาช้อย่อย ๆ ที่  
คอมพิวเตอร์เข้าใจ ทำงานได้ 1  
อย่าง



## บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม (Introduction to Object-Orientation and Abstractions)

### วิธีประมวลผล : ทำอย่างไร

1. เริ่มต้น
2. INITIALIZATION STAGE
  - 2.1 SET TOTAL  $\leftarrow$  0
  - 2.2 SET AVG  $\leftarrow$  0.0
  - 2.3 SET Number  $\leftarrow$  0

1. INPUT / READ / ENTER
2. ....
3. ....
4. ....
5. ...
6. ....
7. สิ้นสุด/ จบ / เสร็จสิ้น

Initialization Stage

Initialization Stage  
Input Stage  
Process Stage  
Output Stage

เขียนมาข้อย่อย ๆ ที่คอมพิวเตอร์เข้าใจ  
ทำงานได้ 1 อย่าง



## บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม (Introduction to Object-Orientation and Abstractions)

### วิธีประมวลผล : ทำอย่างไร

1. เริ่มต้น
2. INITIALIZATION STAGE
  - 2.1 SET TOTAL  $\leftarrow$  0
  - 2.2 SET AVG  $\leftarrow$  0.0
  - 2.3 SET Number  $\leftarrow$  0
3. PRINT "Enter Number of Student : "
4. READ Number
  1. ....
  2. ....
  3. ....
  4. ...
  5. ....
  6. สิ้นสุด/ จบ / เสร็จสิ้น

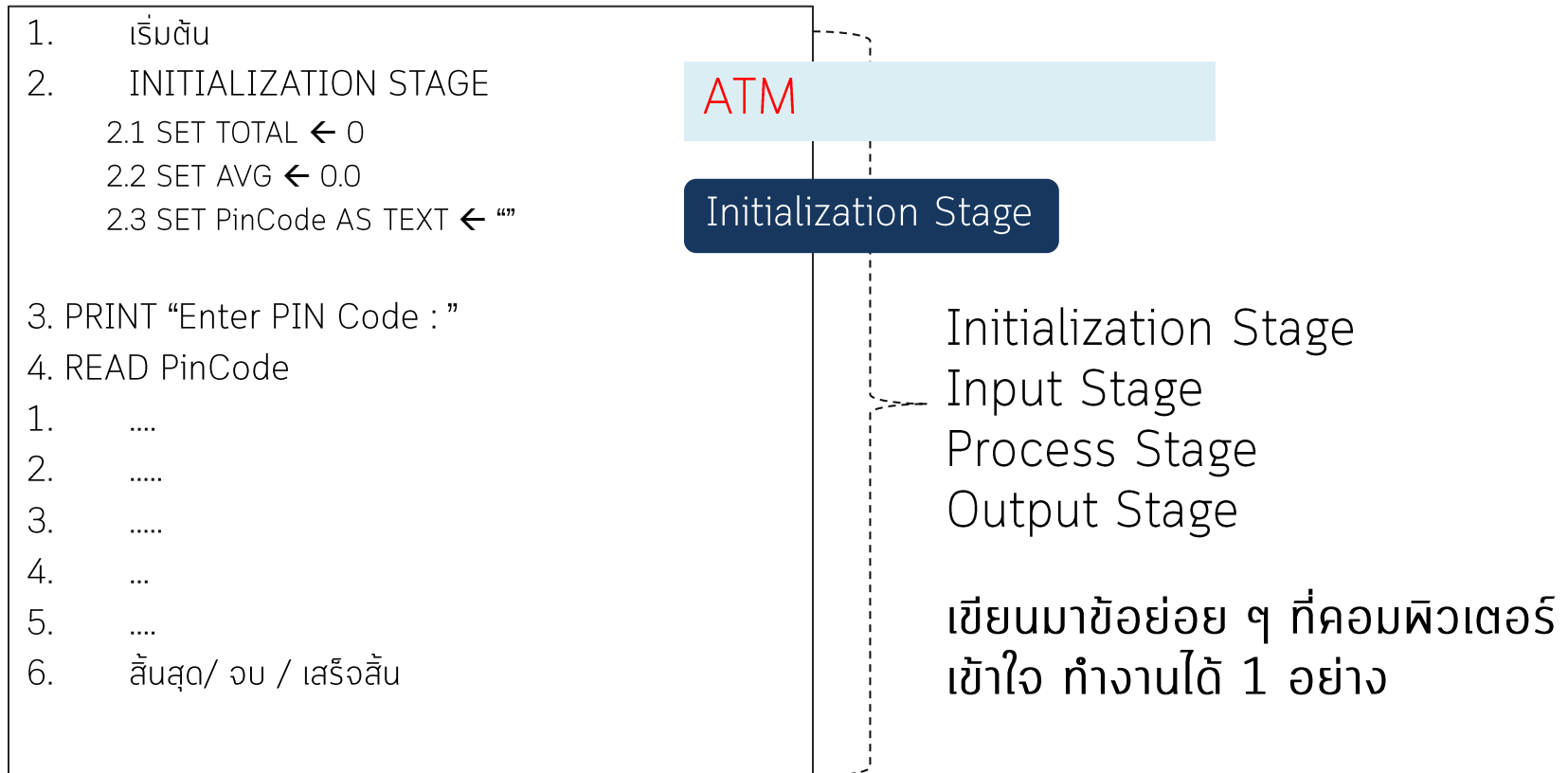
Initialization Stage

Initialization Stage  
Input Stage  
Process Stage  
Output Stage

เขียนมาช้อย่อย ๆ ที่คอมพิวเตอร์  
เข้าใจ ทำงานได้ 1 อย่าง

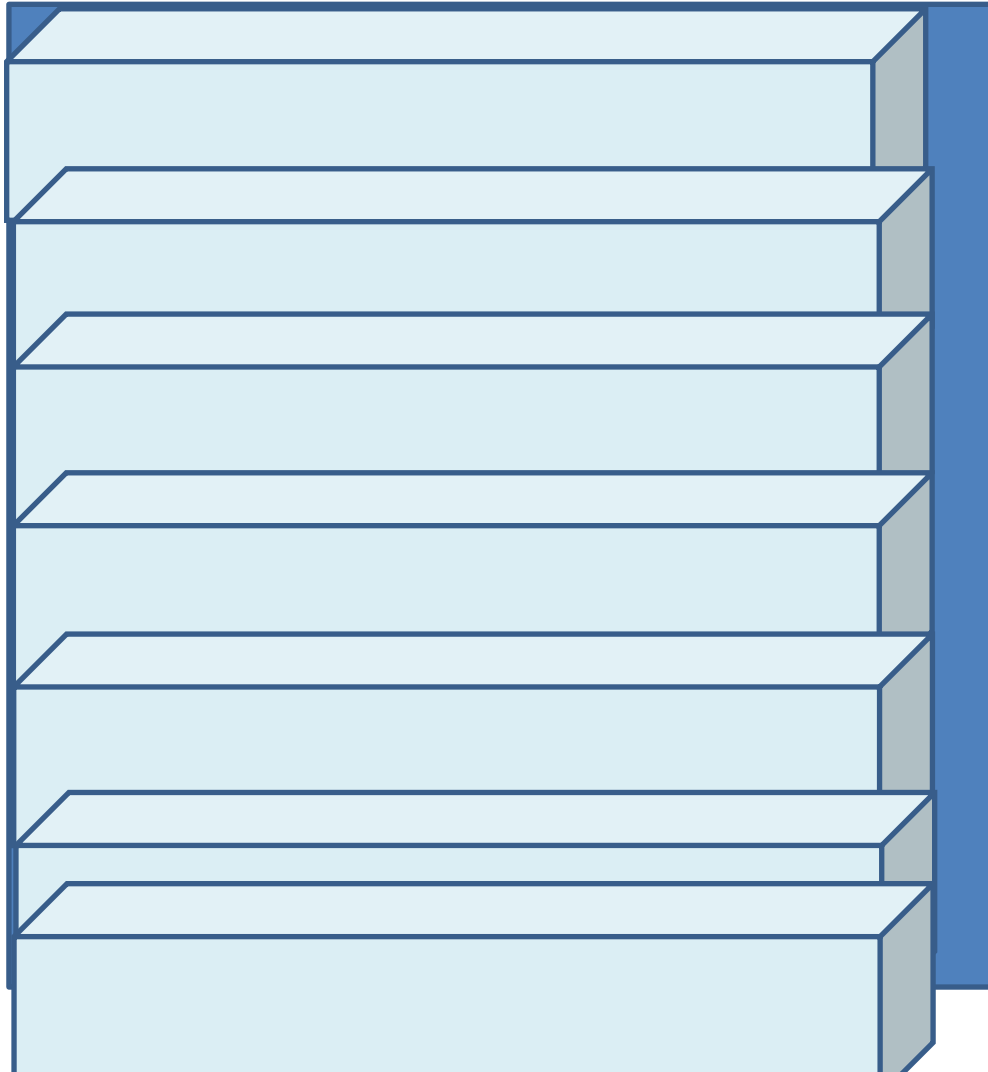
## บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม (Introduction to Object-Orientation and Abstractions)

### วิธีประมวลผล : ทำอย่างไร



จะจัดกล่องเข้าในห้องที่มีขนาด 10\*20 ได้กี่กล่อง

10 M



กว้าง :  เมตร

ยาว :  เมตร

ผลลัพธ์ :  กล่อง

คำนวณ

ยกเลิก

20 M

$k = 2 \dots 3$

$y = 5 \dots 9$

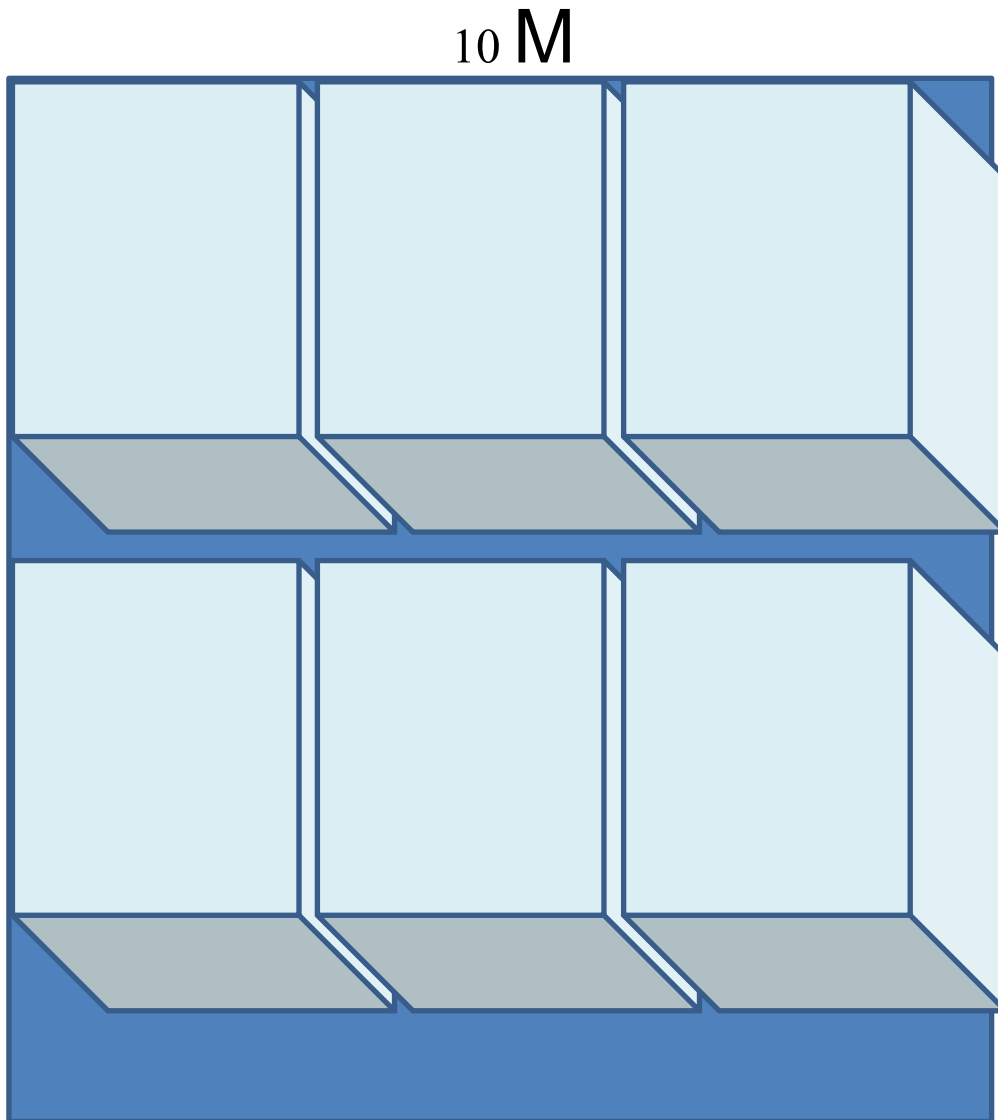
มากที่สุดได้ กี่กล่อง

ยาว :

กว้าง :



จะจัดกล่องเข้าในห้องที่มีขนาด 10\*20 ได้กี่  
กล่อง



กว้าง :  เมตร

ยาว :  เมตร

ผลลัพธ์ :  กล่อง

20 M

$k = 2 \dots\dots 3$

$y = 5 \dots\dots 9$

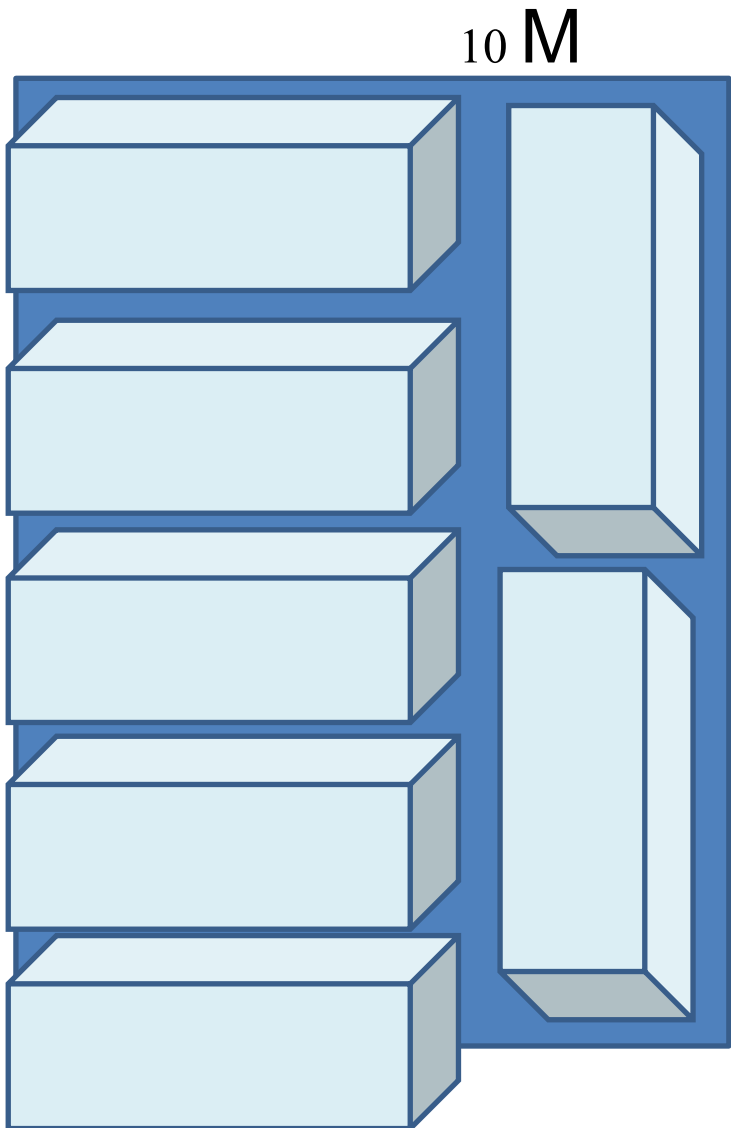
มากที่สุดได้ กี่กล่อง

ยาว :



กว้าง :

จะจัดกล่องเข้าในห้องที่มีขนาด 10\*20 ได้กี่กล่อง



กว้าง :  เมตร

ยาว :  เมตร

ผลลัพธ์ :  กล่อง

20 M

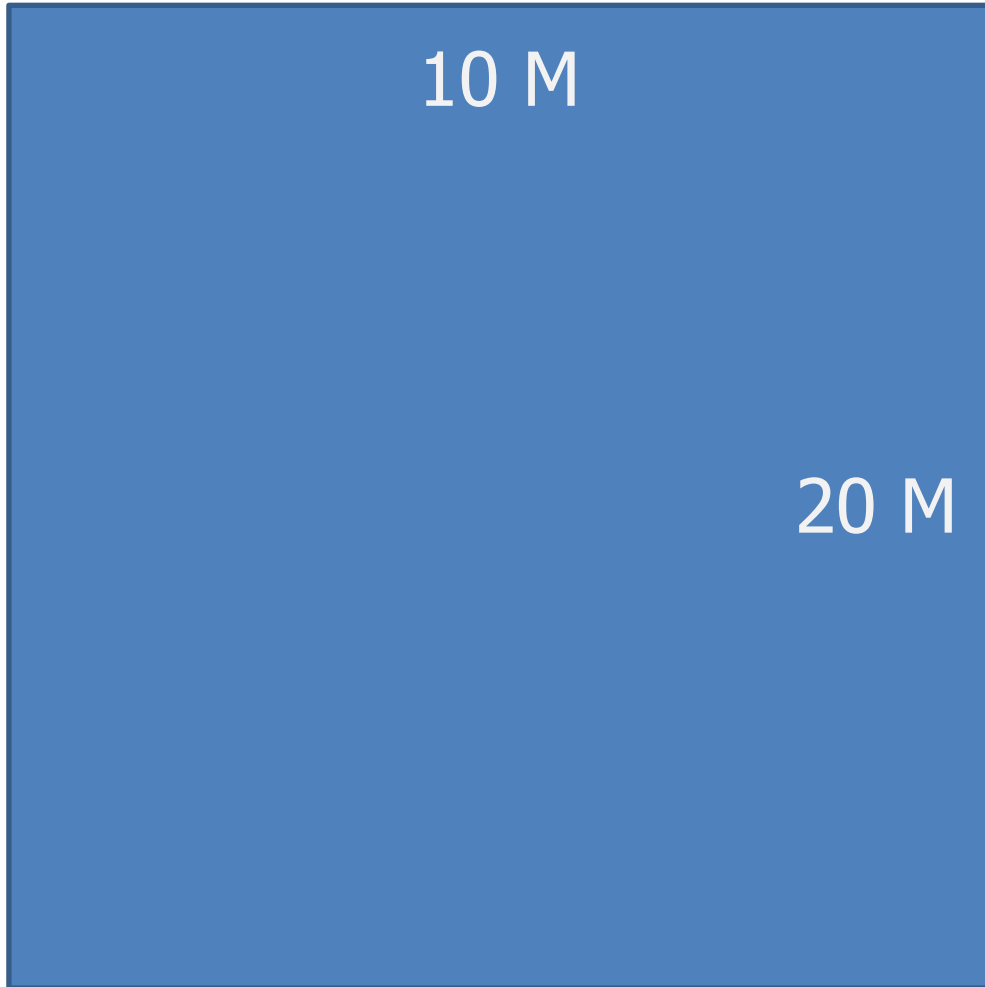
$k = 2 \dots\dots 3 \quad 4$   
 $y = 5 \dots\dots 9 \quad 7$   
มากที่สุดได้ กี่กล่อง

ยาว :

กว้าง :



# Algorithm ?



กว้าง :	<input type="text" value="X"/>	เมตร
ยาว :	<input type="text" value="X"/>	เมตร
ผลลัพธ์ :	<input type="text" value="X"/>	กล่อง

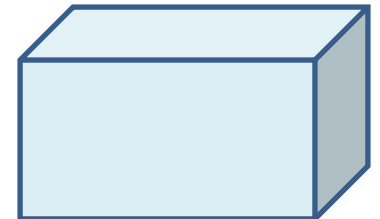
จะจัดกล่องเข้าในห้องที่มีขนาด  
 $10 \times 20$  ได้กี่กล่อง

$$ก = 2 \dots\dots 3$$

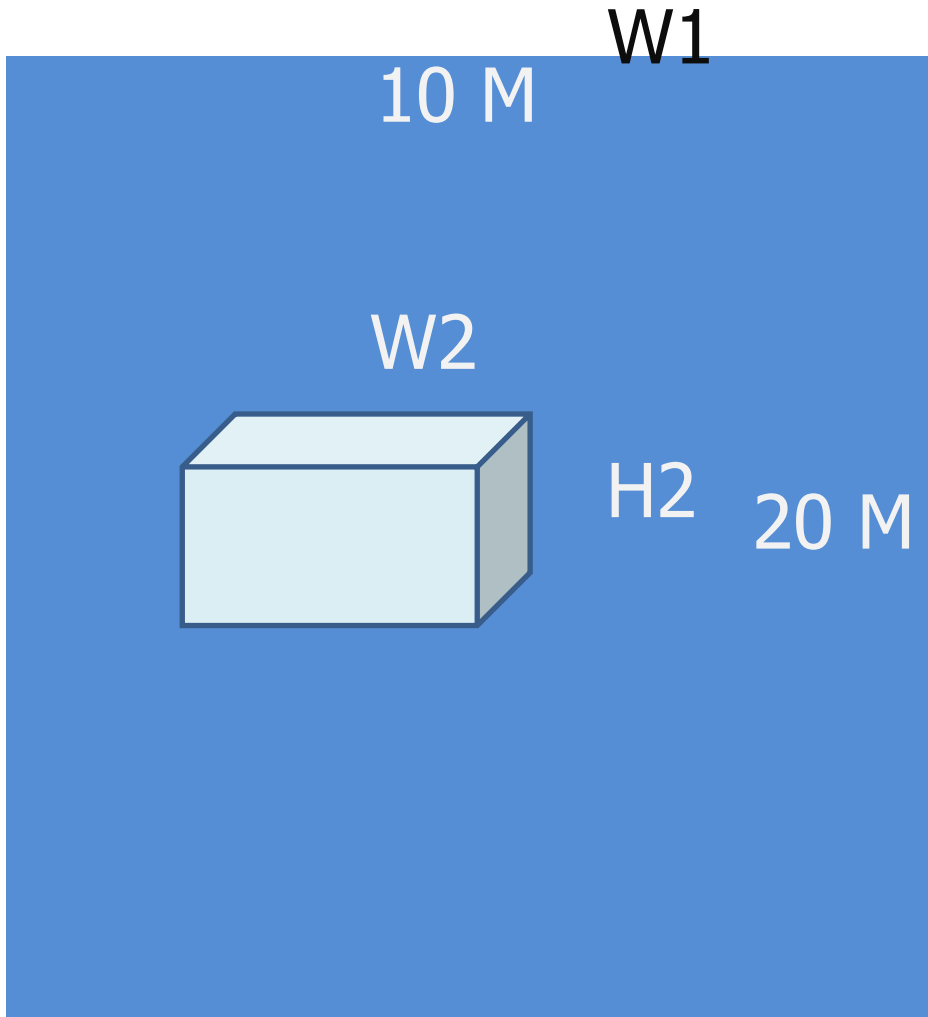
$$ย = 5 \dots\dots 9$$

มากที่สุดได้ กี่กล่อง

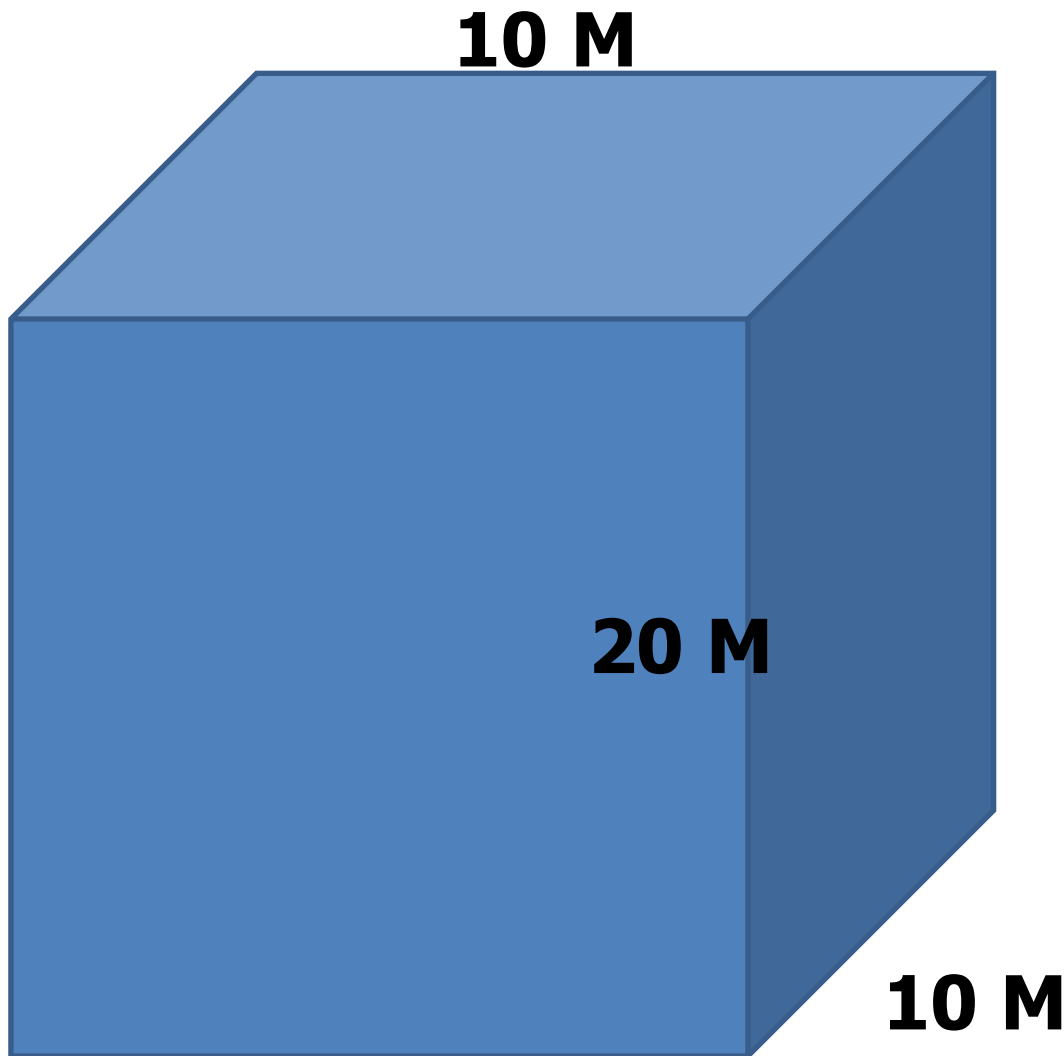
ยาว :



กว้าง :



```
If w1<=H1 THEN
    If (w2<w1) AND (H2<H1) THEN
        ELSE
            END IF
    ELSE
        PRINT "Invalid INPUT"
    END IF
```



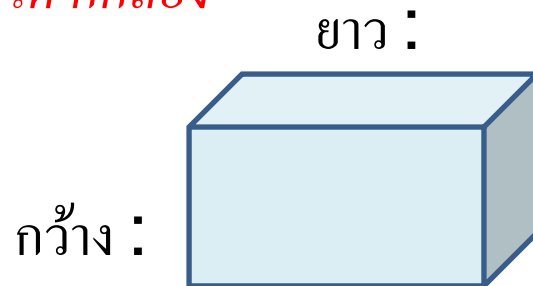
กว้าง :	<input type="text" value="X"/>	เมตร
ยาว :	<input type="text" value="X"/>	เมตร
ผลลัพธ์ :	<input type="text" value="X"/>	กล่อง

จะจัดกล่องเข้าในห้องที่มีขนาด  
10\*20 ได้กี่กล่อง

$$ก = 2 \dots\dots 3$$

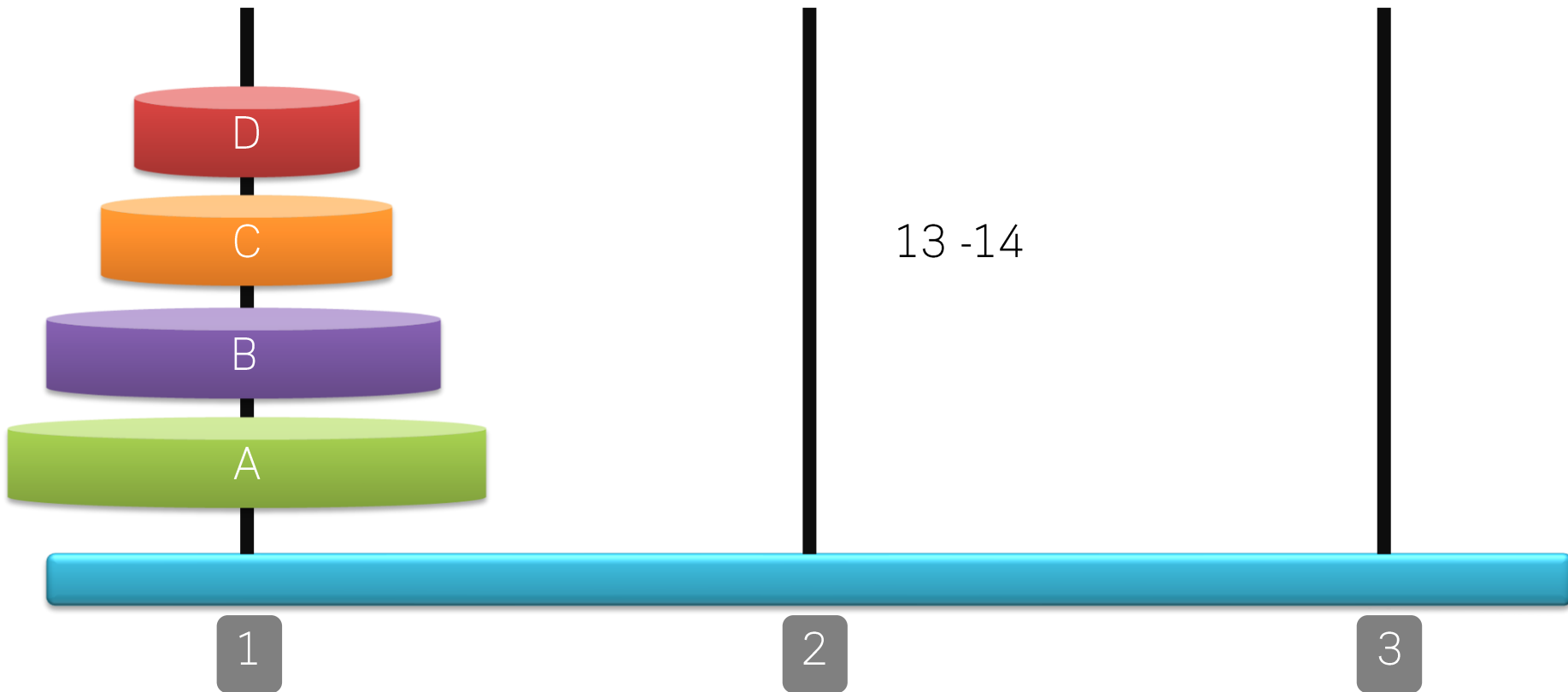
$$ย = 5 \dots\dots 9$$

มากที่สุด ได้ กี่กล่อง





บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

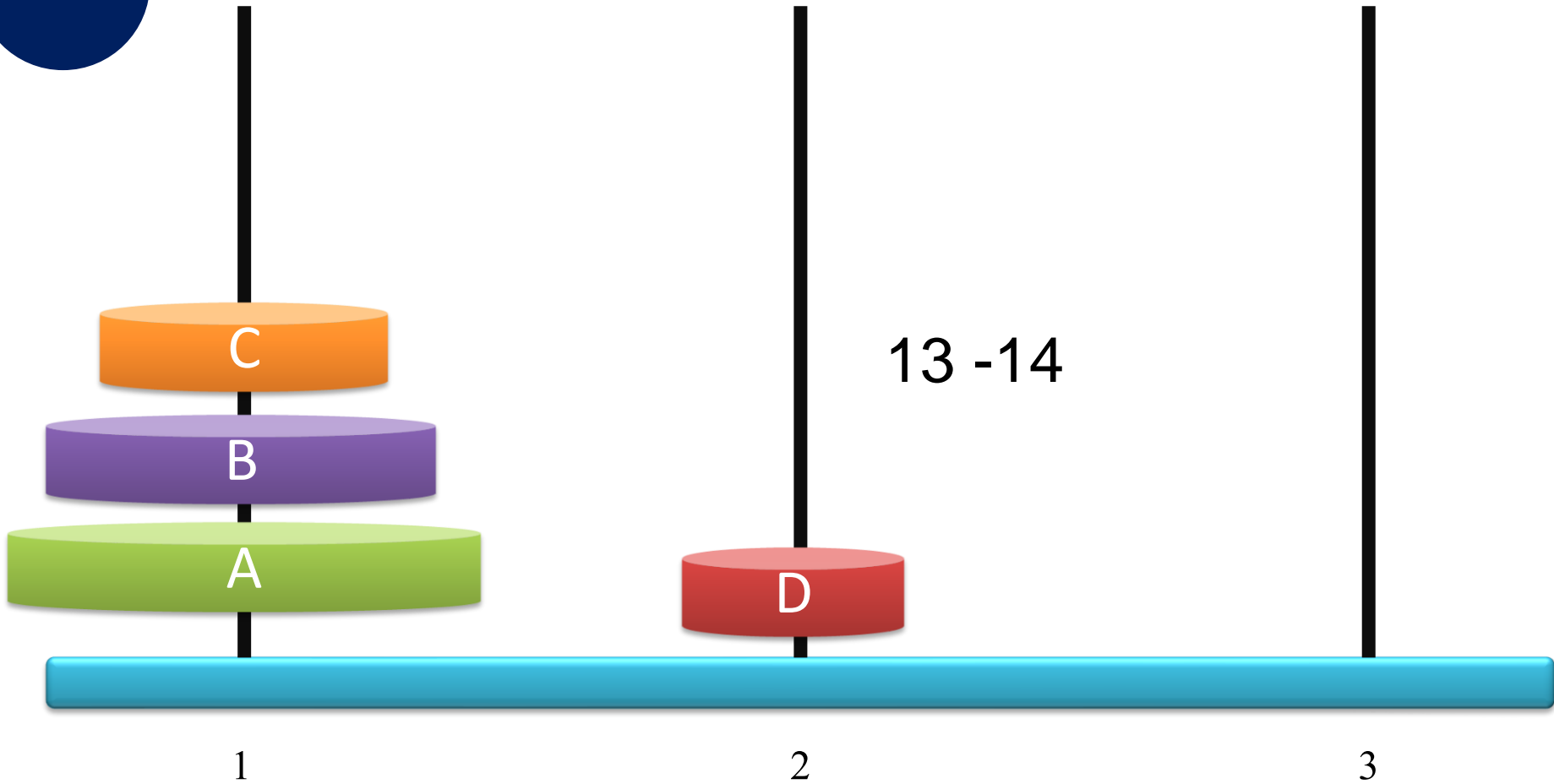


Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- 2. ย้ายได้ครั้งละ 1 แผ่น
- 3. ห้ามแผ่นใหญ่กับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

1

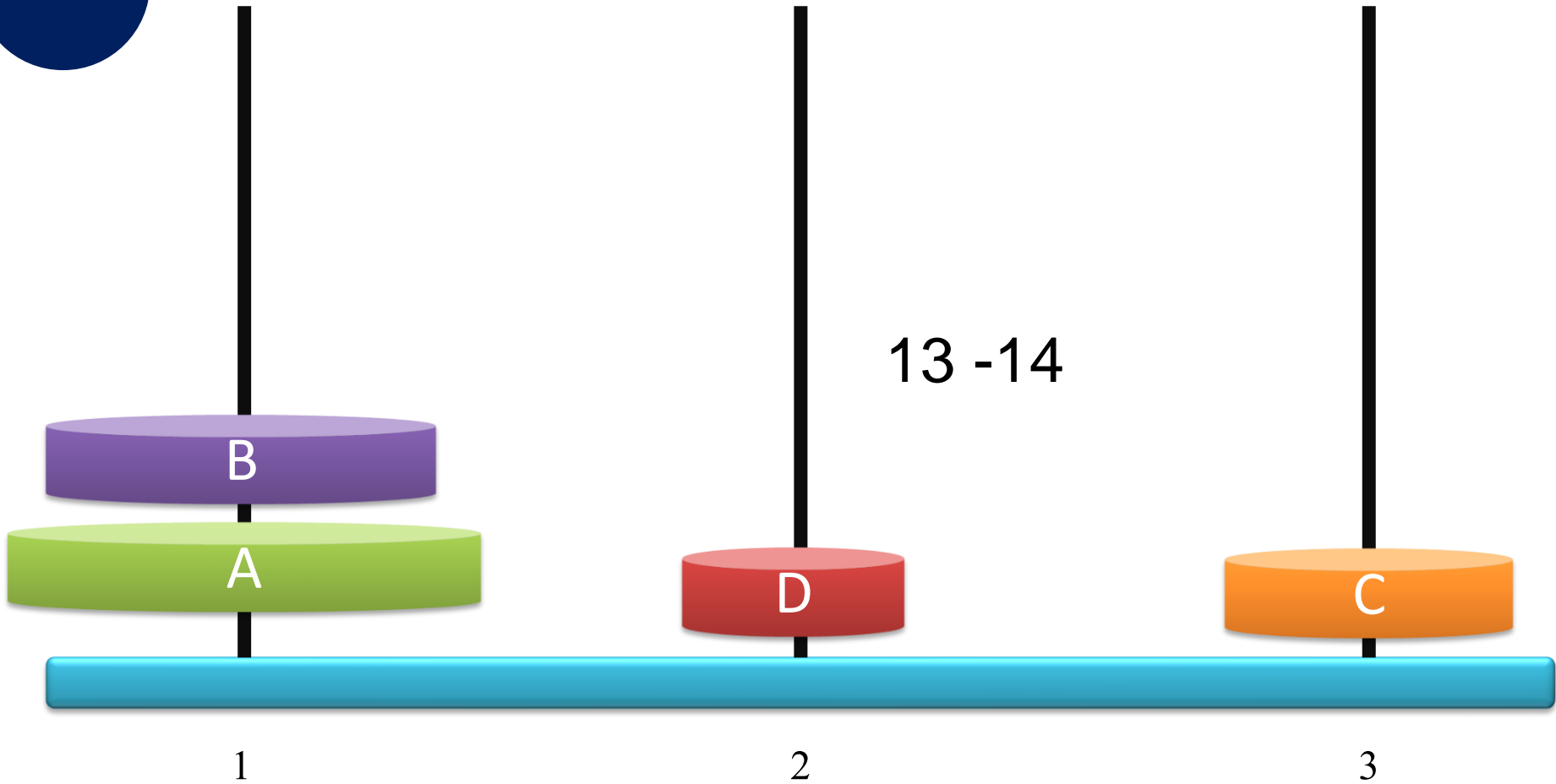


## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

2

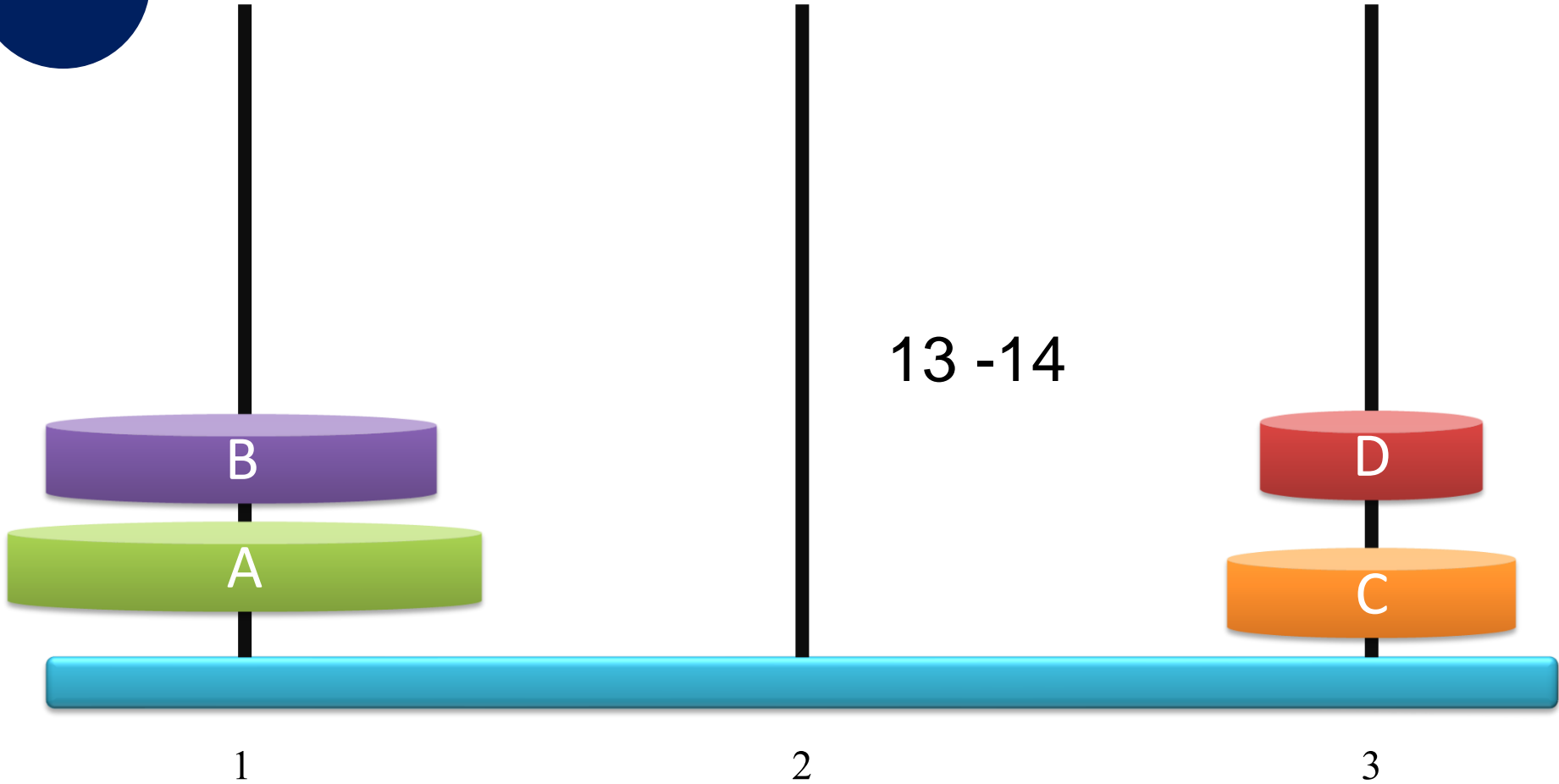


## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
2. ย้ายได้ครั้งละ 1 แผ่น
3. ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

3

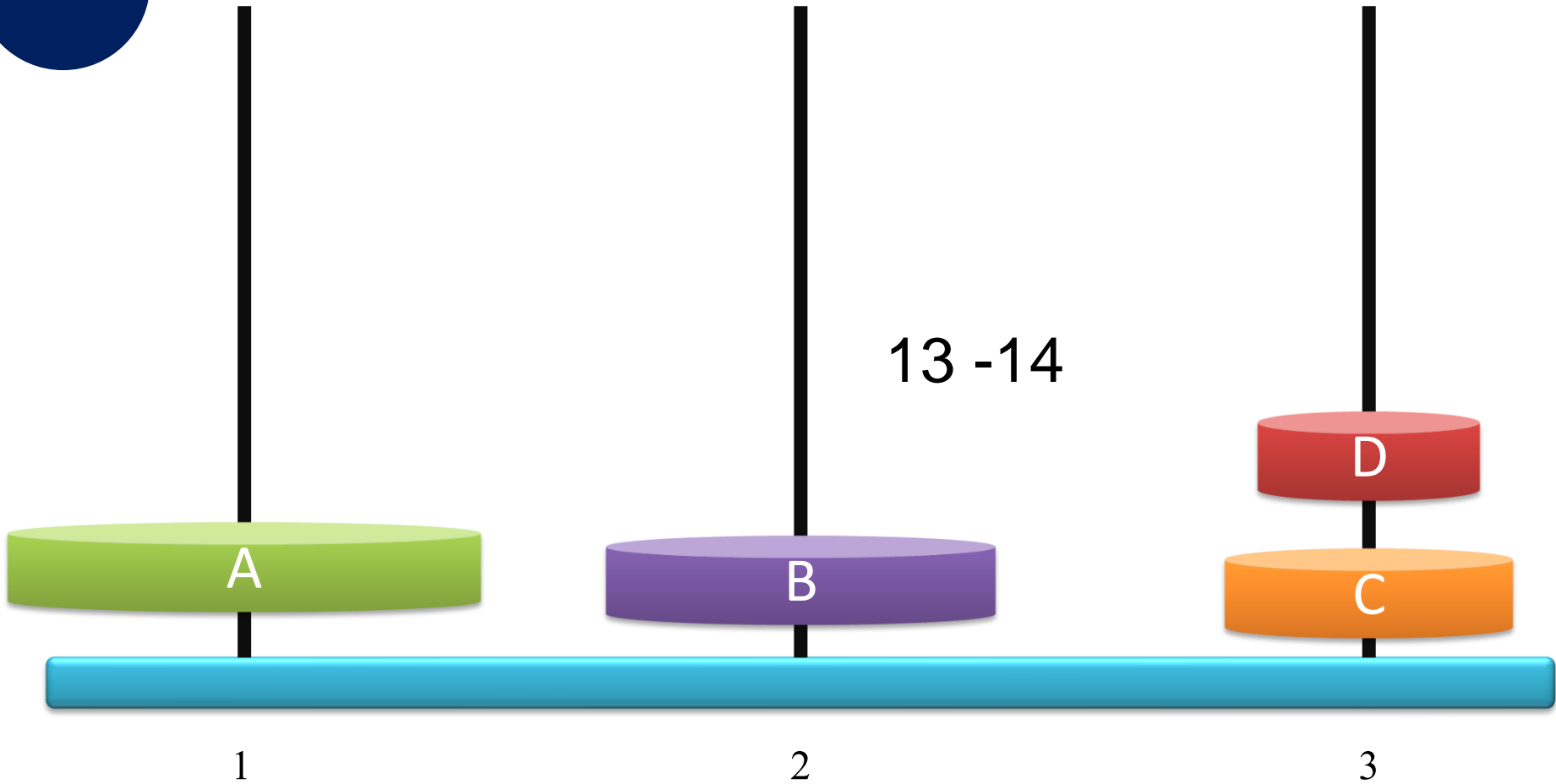


## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

4

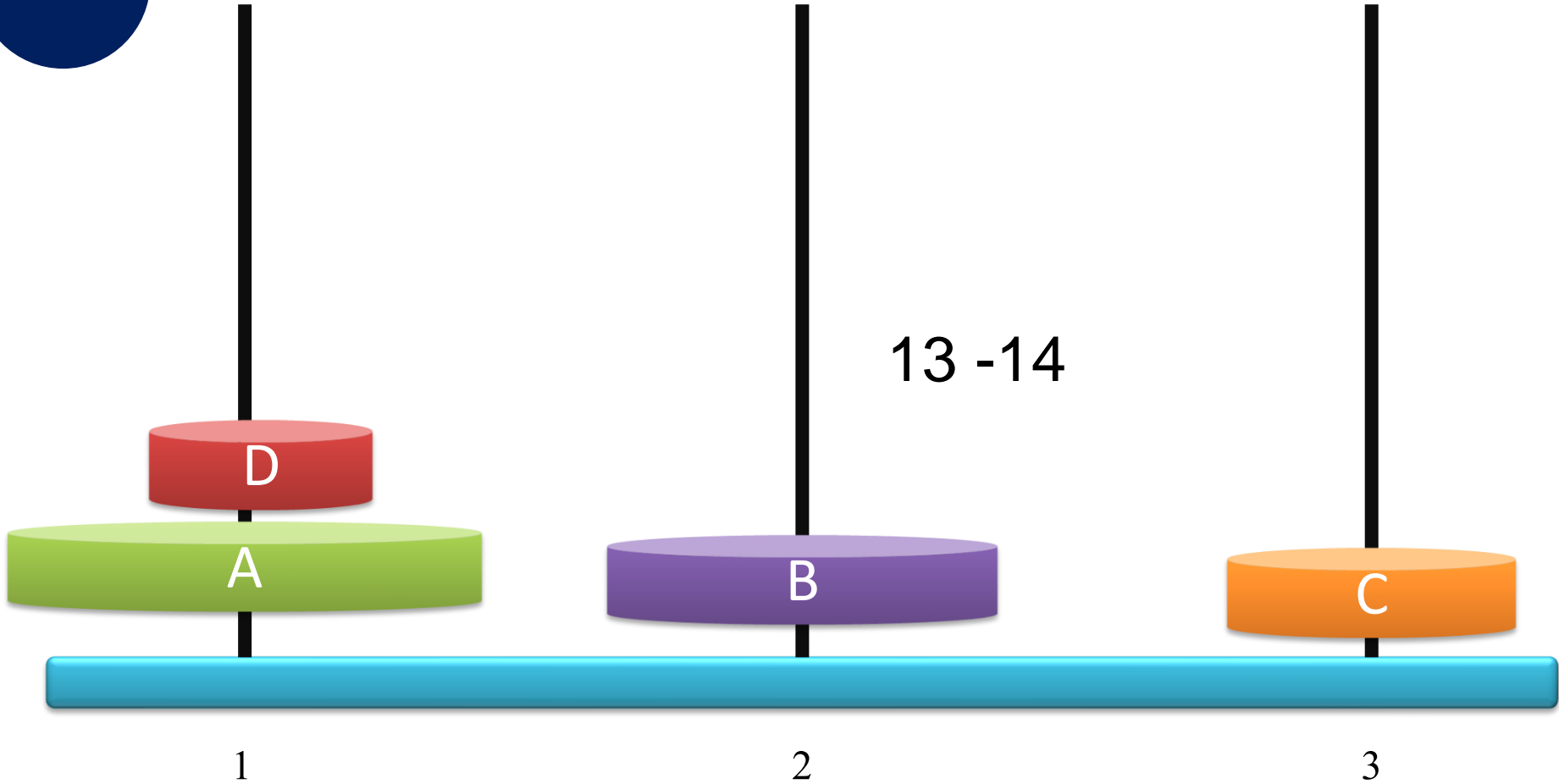


## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

5

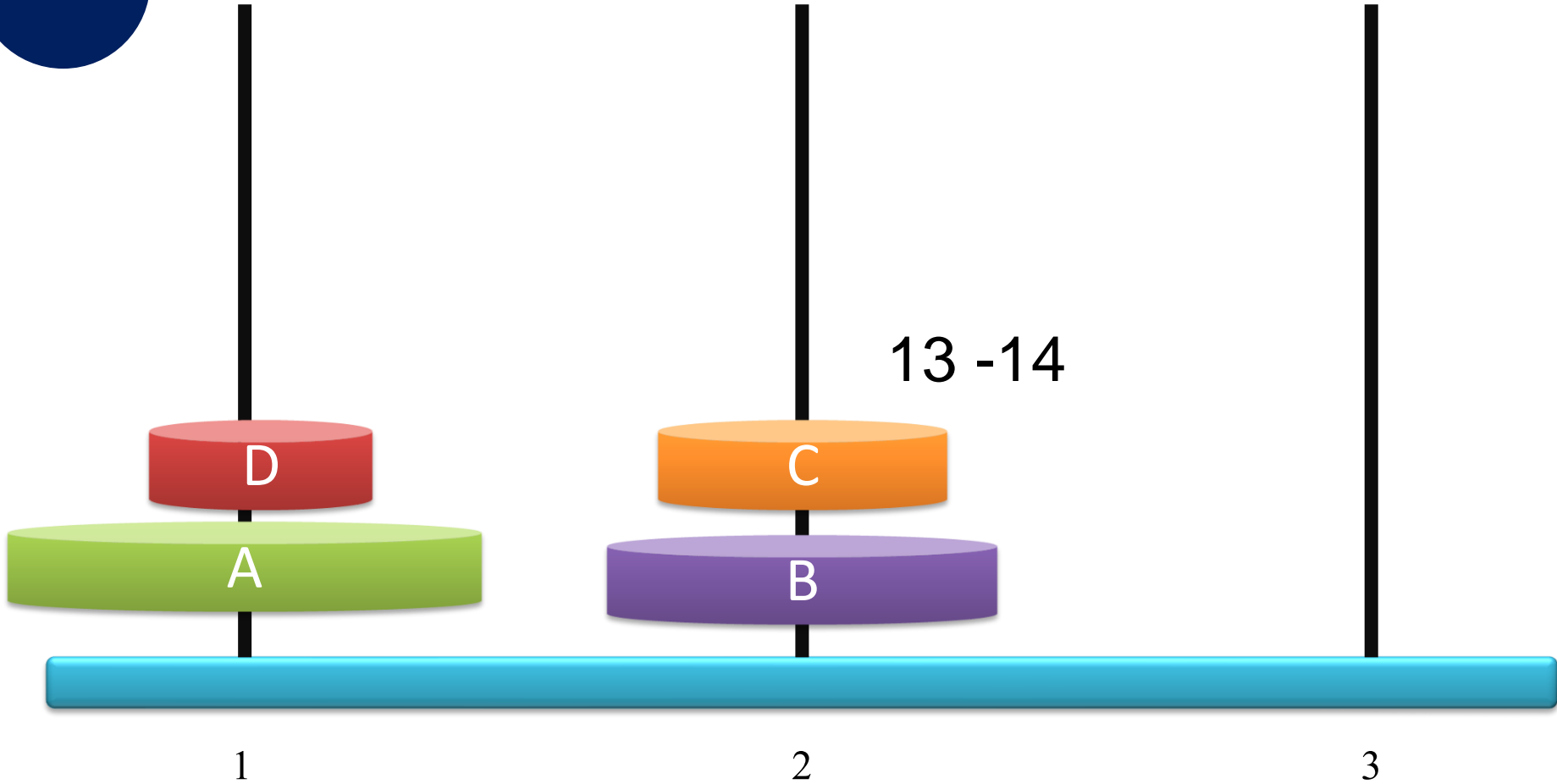


## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

6

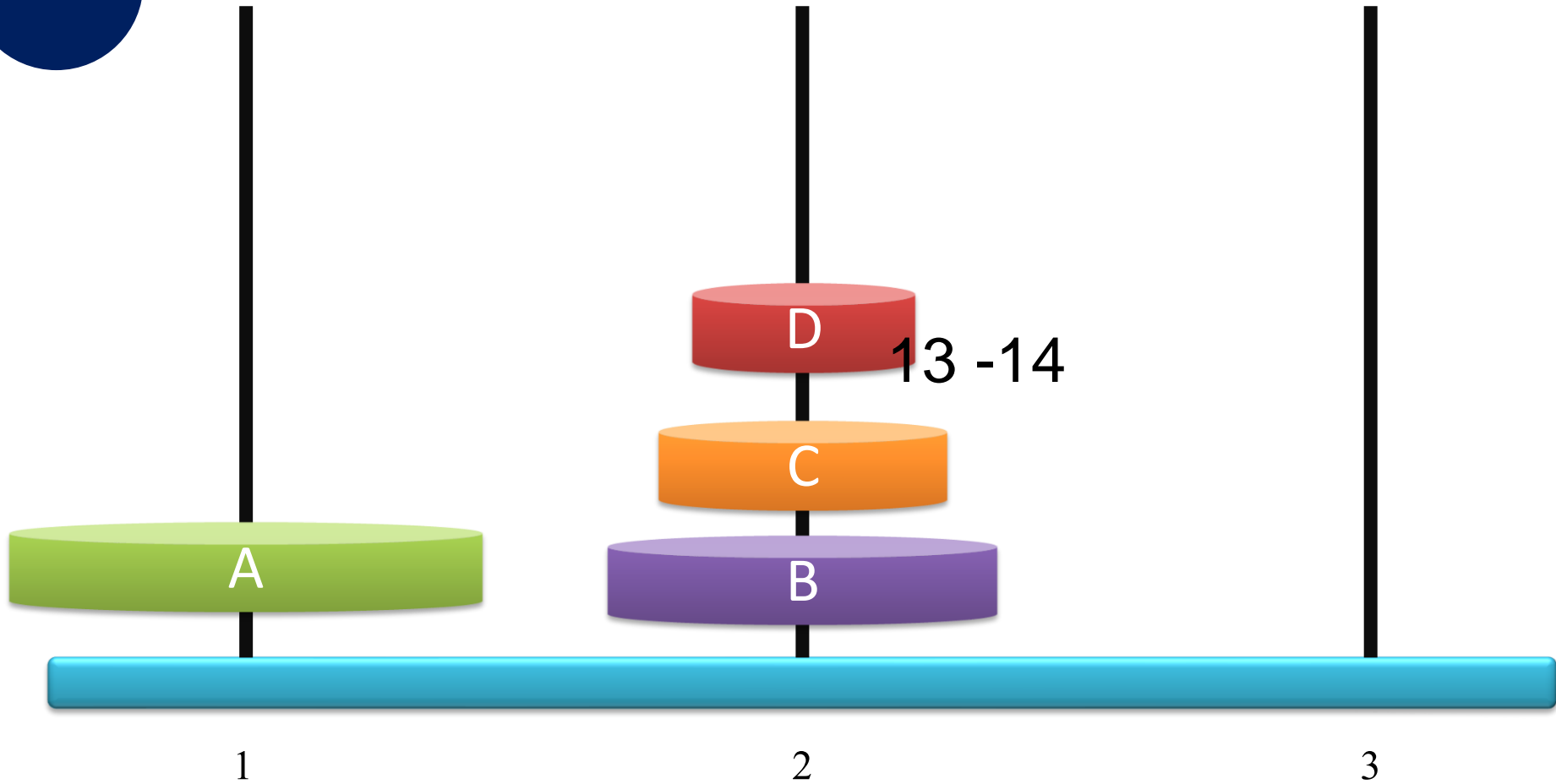


## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

7

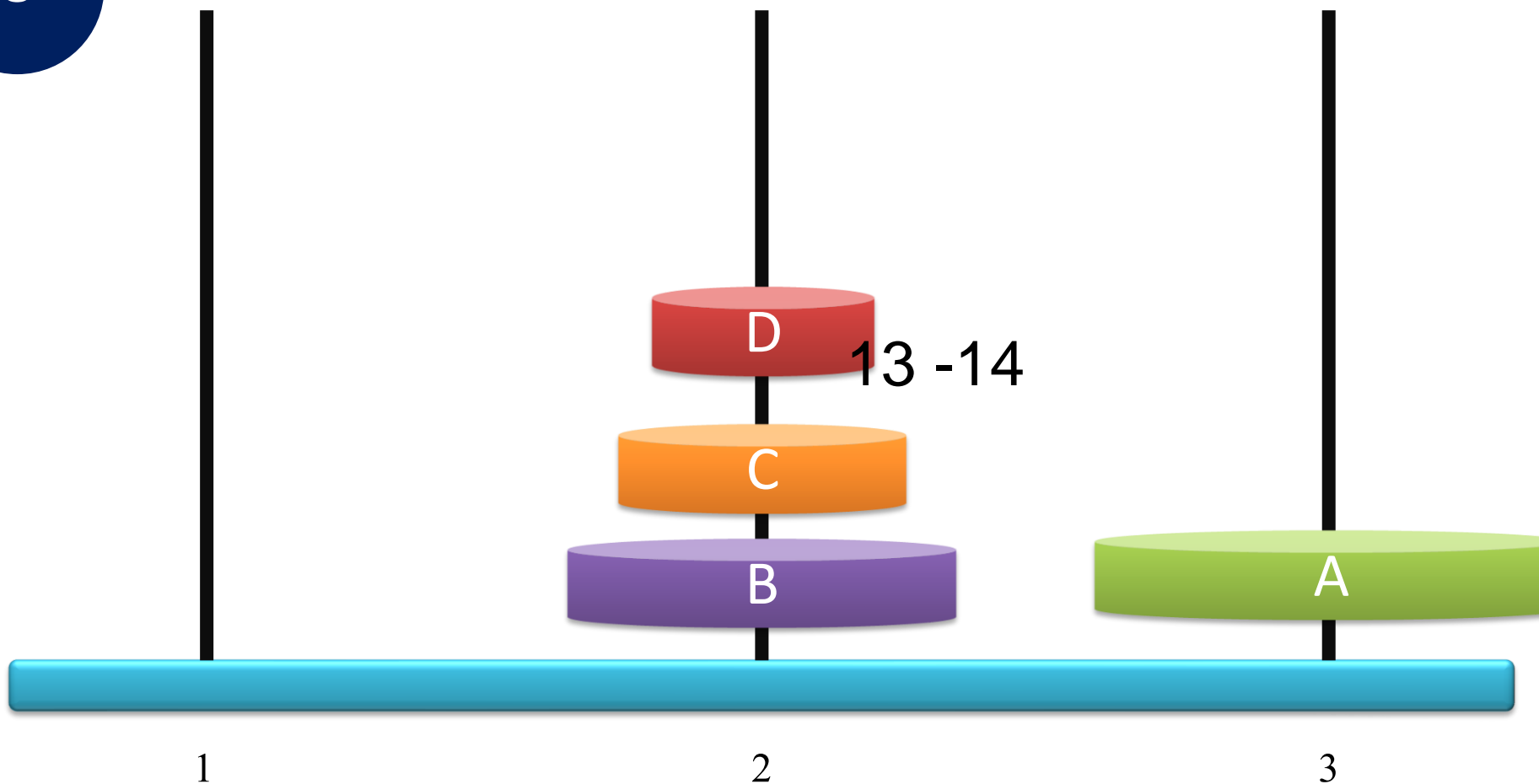


## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้จำนวนครั้งน้อยที่สุด ชนะ*

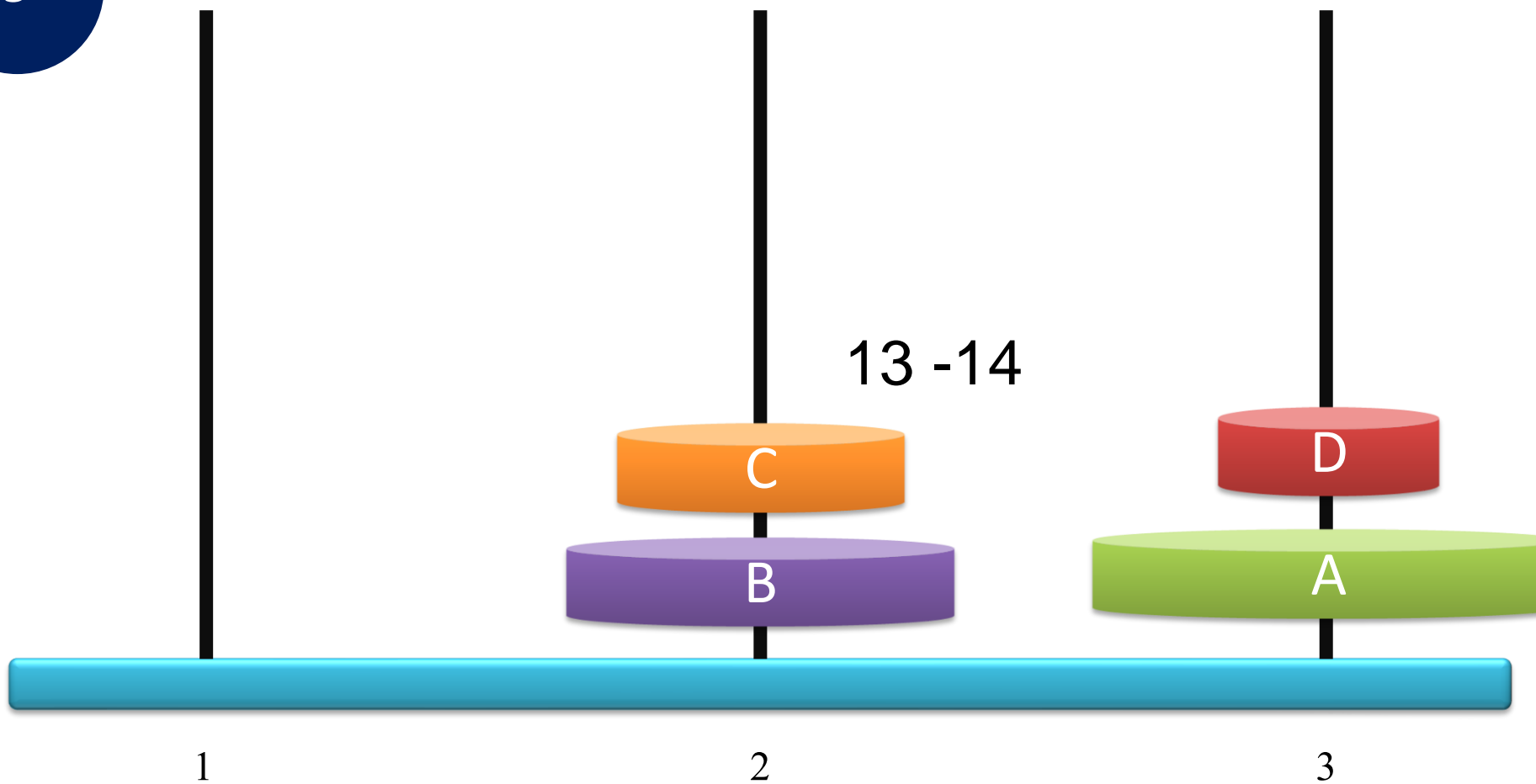




## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

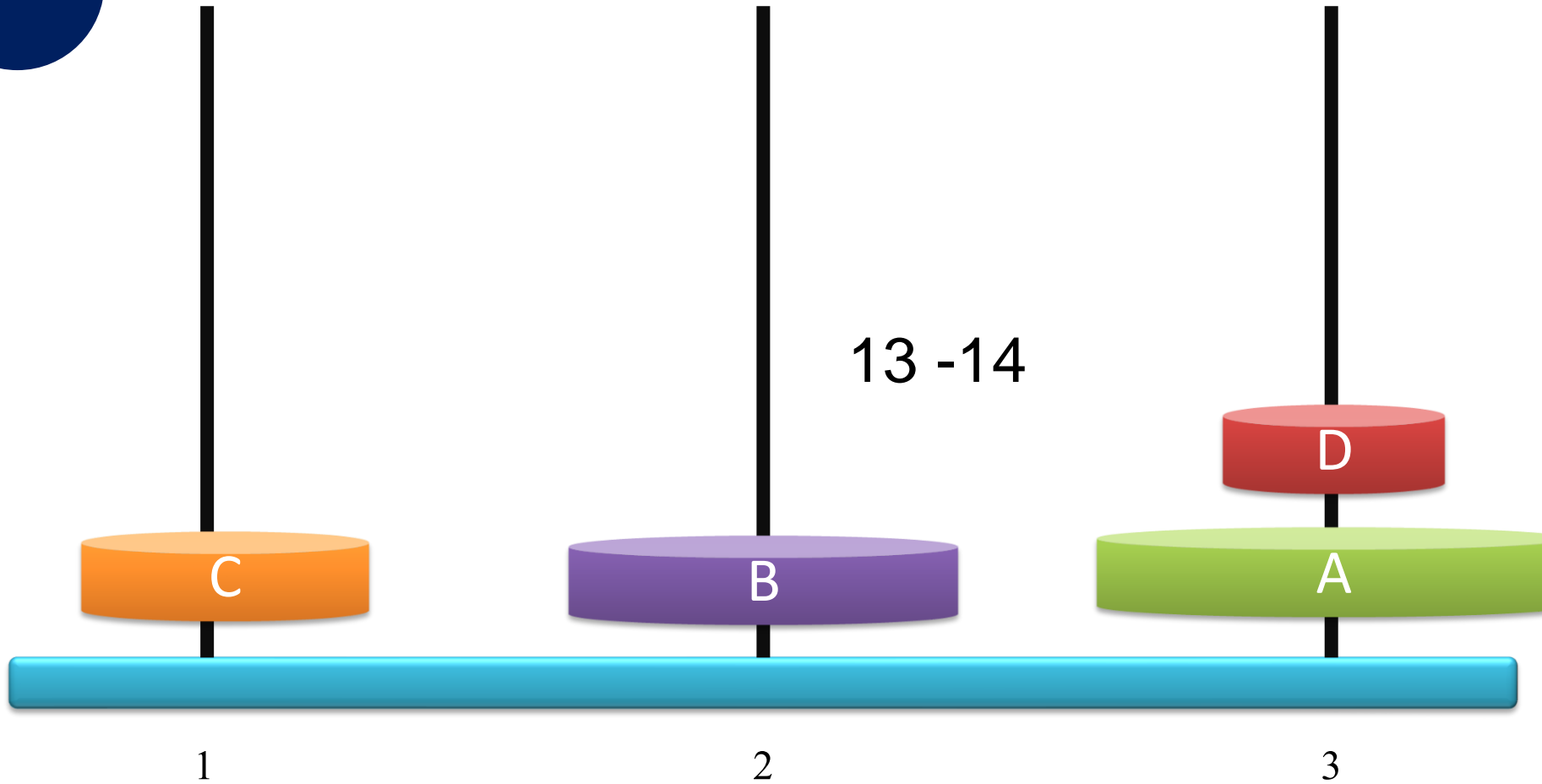


## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

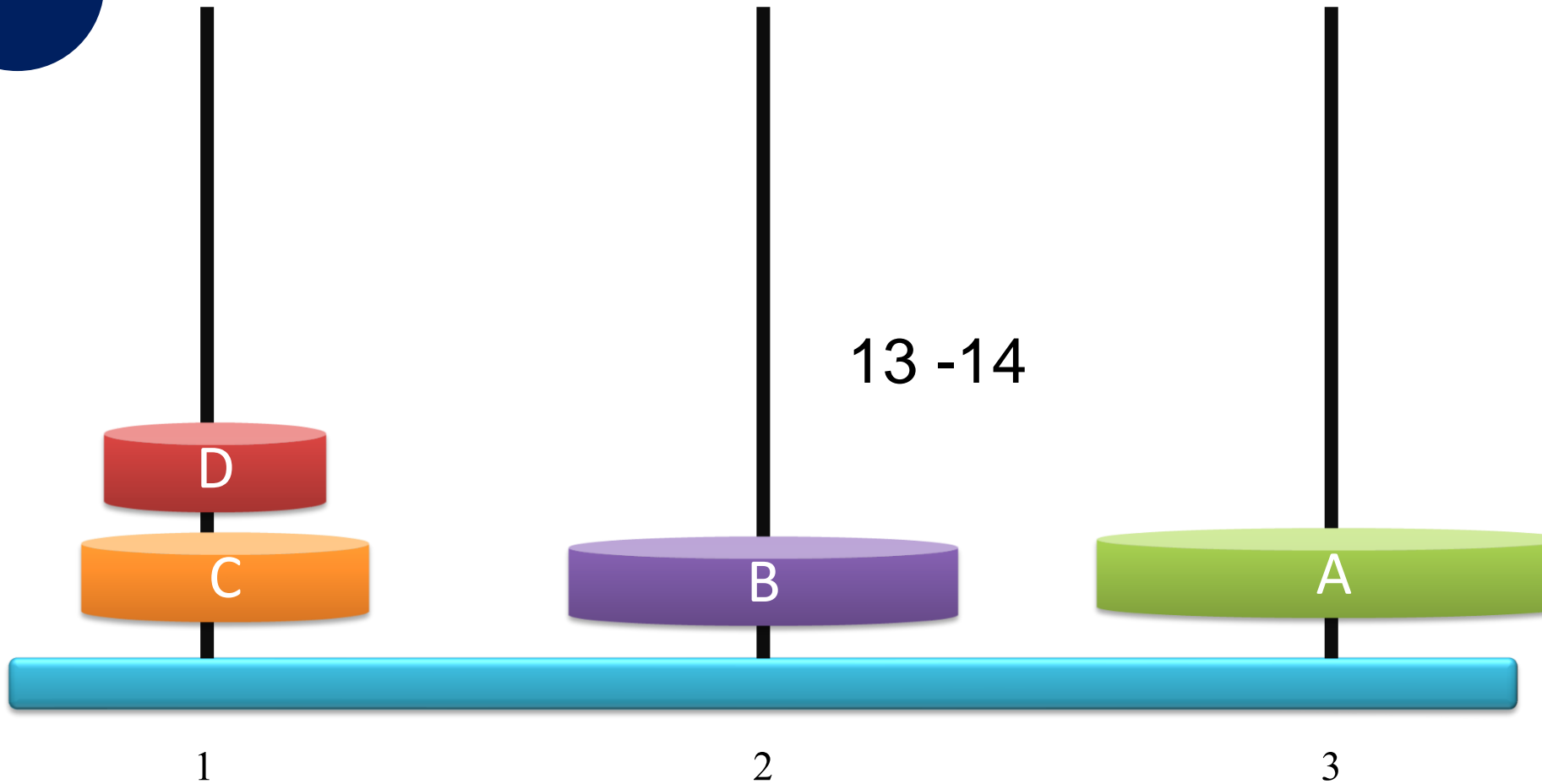
10



## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

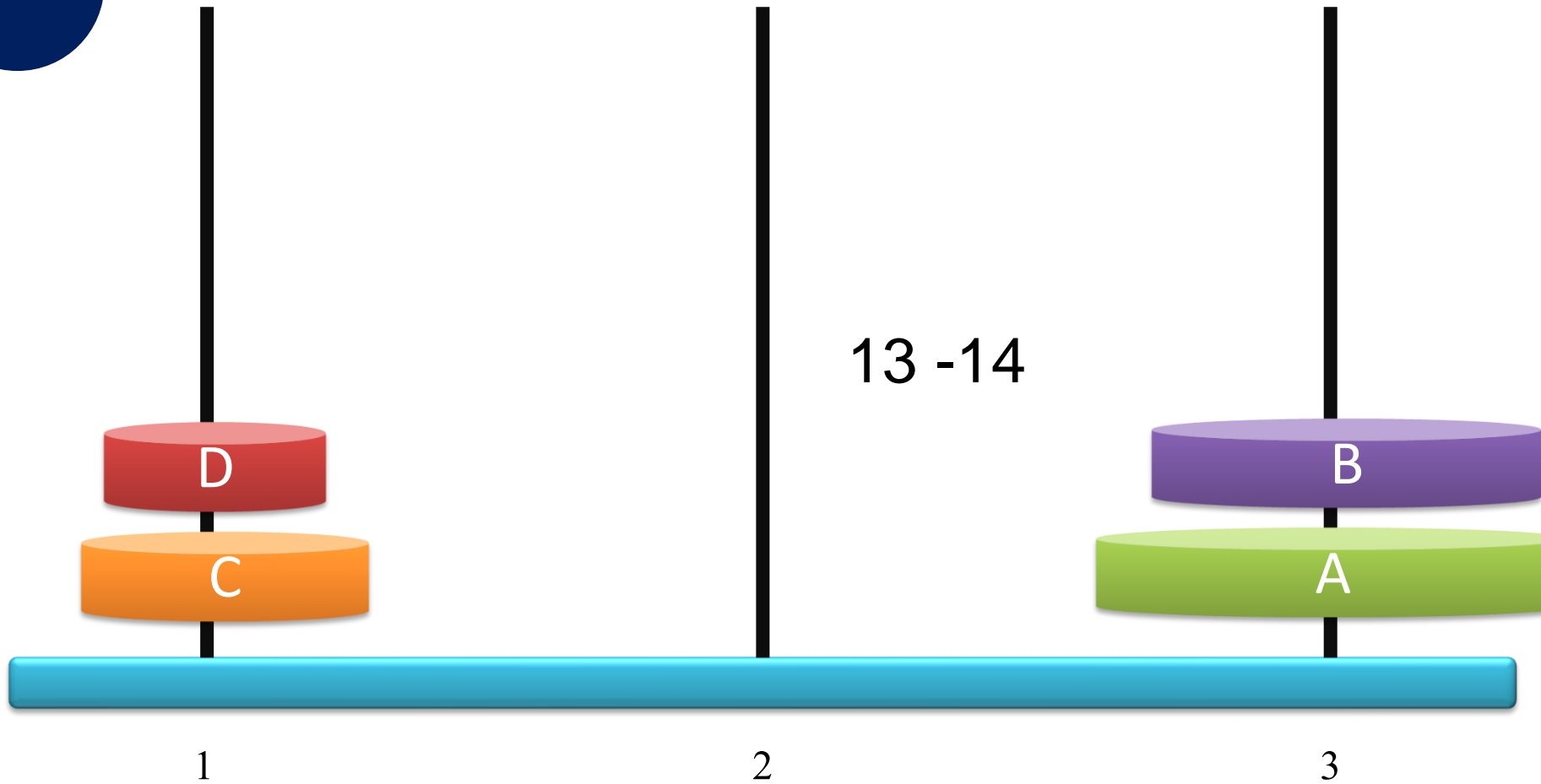
*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*



## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

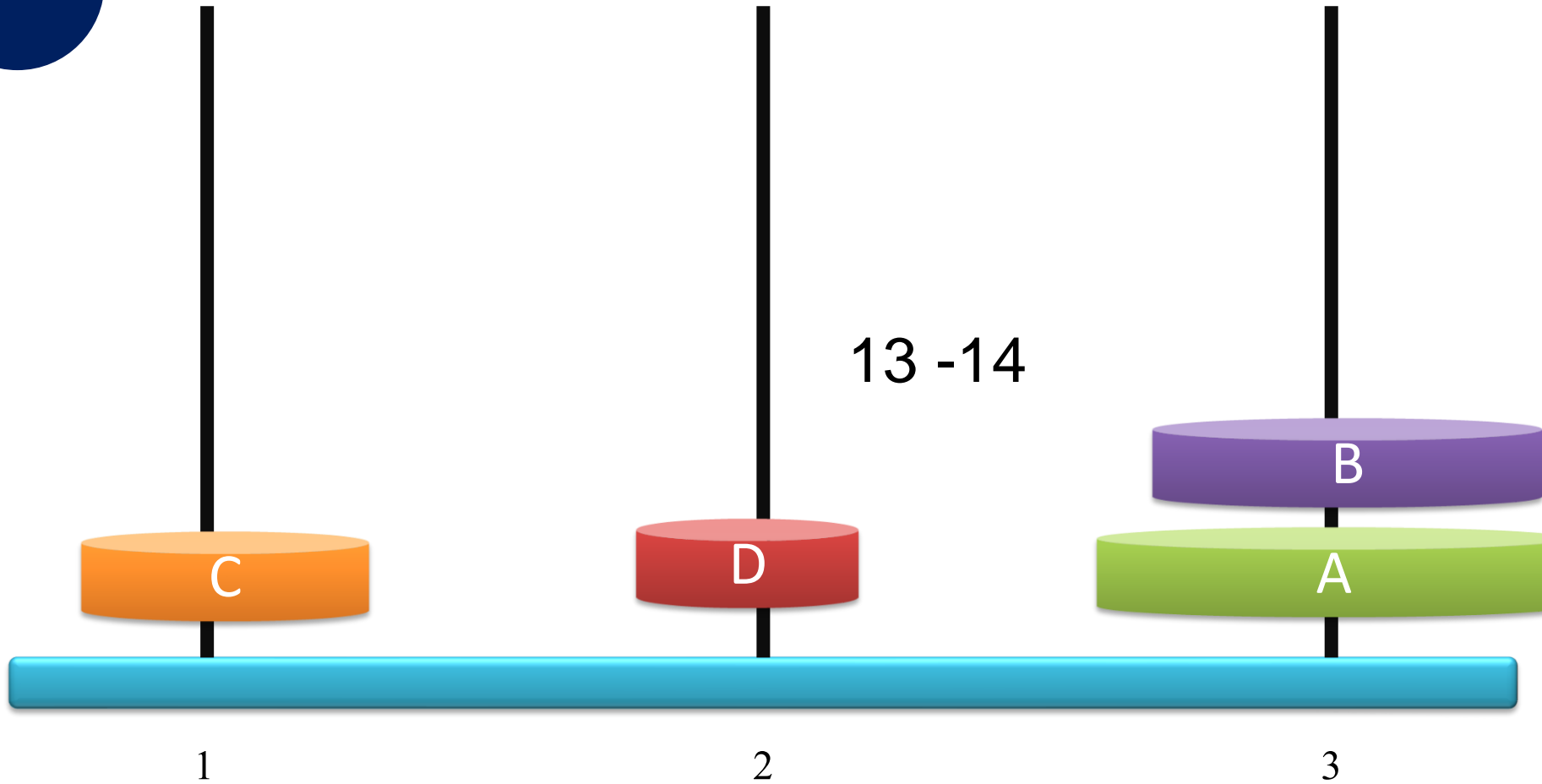


## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

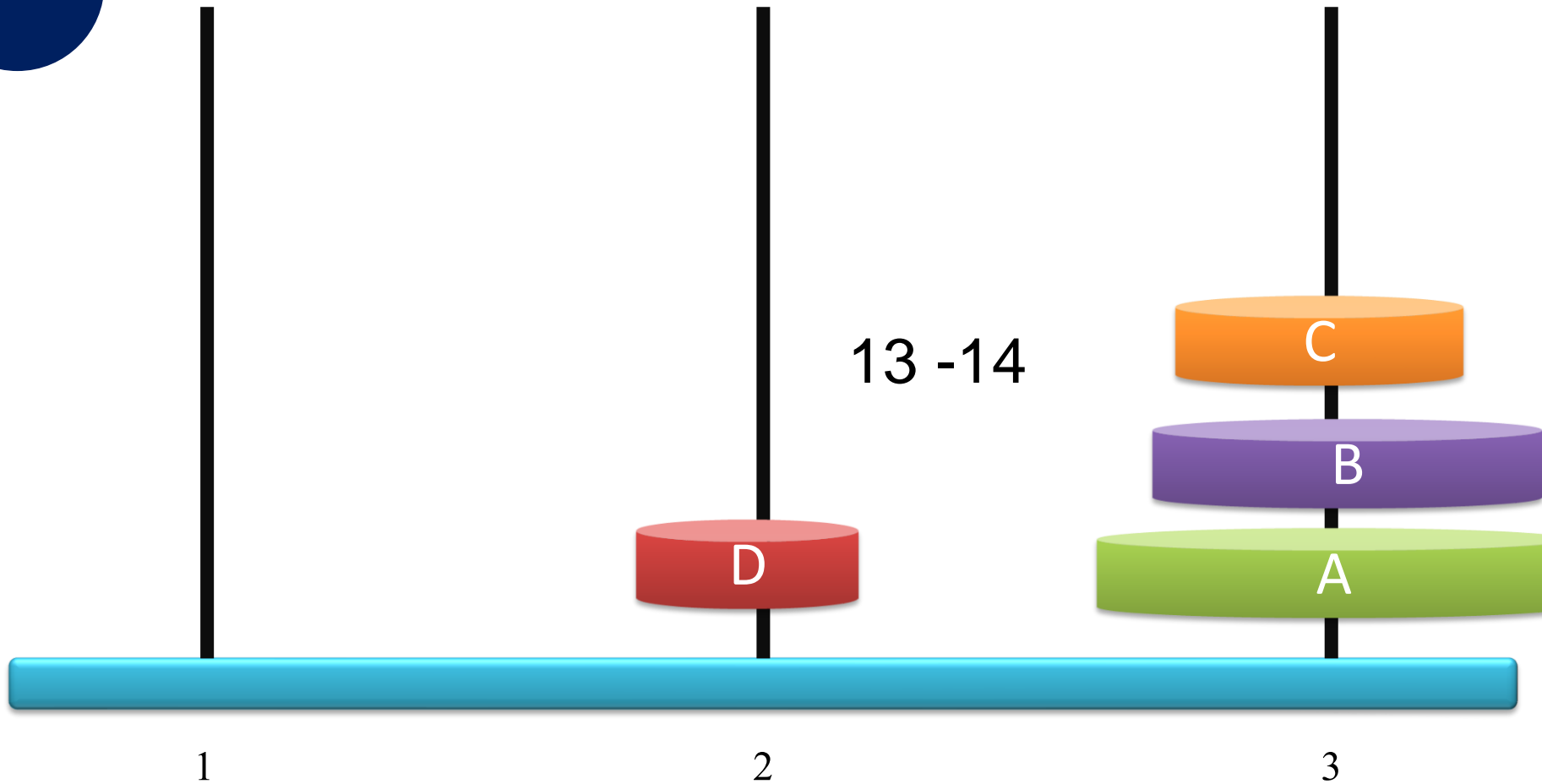
13



## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
2. ย้ายได้ครั้งละ 1 แผ่น
3. ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

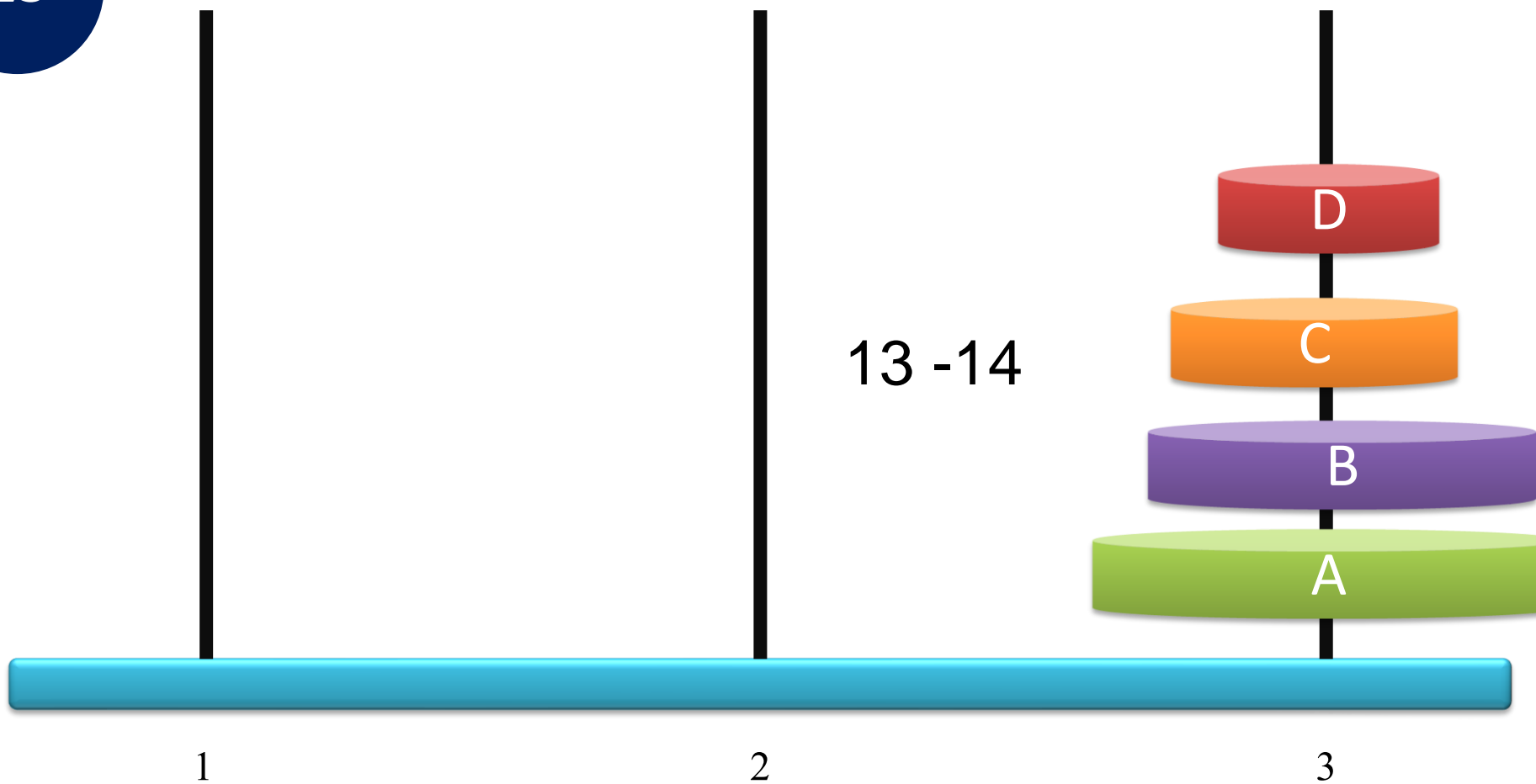


## Tower of Hanoi

- กติกา 1. ให้ย้ายจาก 1-3
- ย้ายได้ครั้งละ 1 แผ่น
- ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*

15



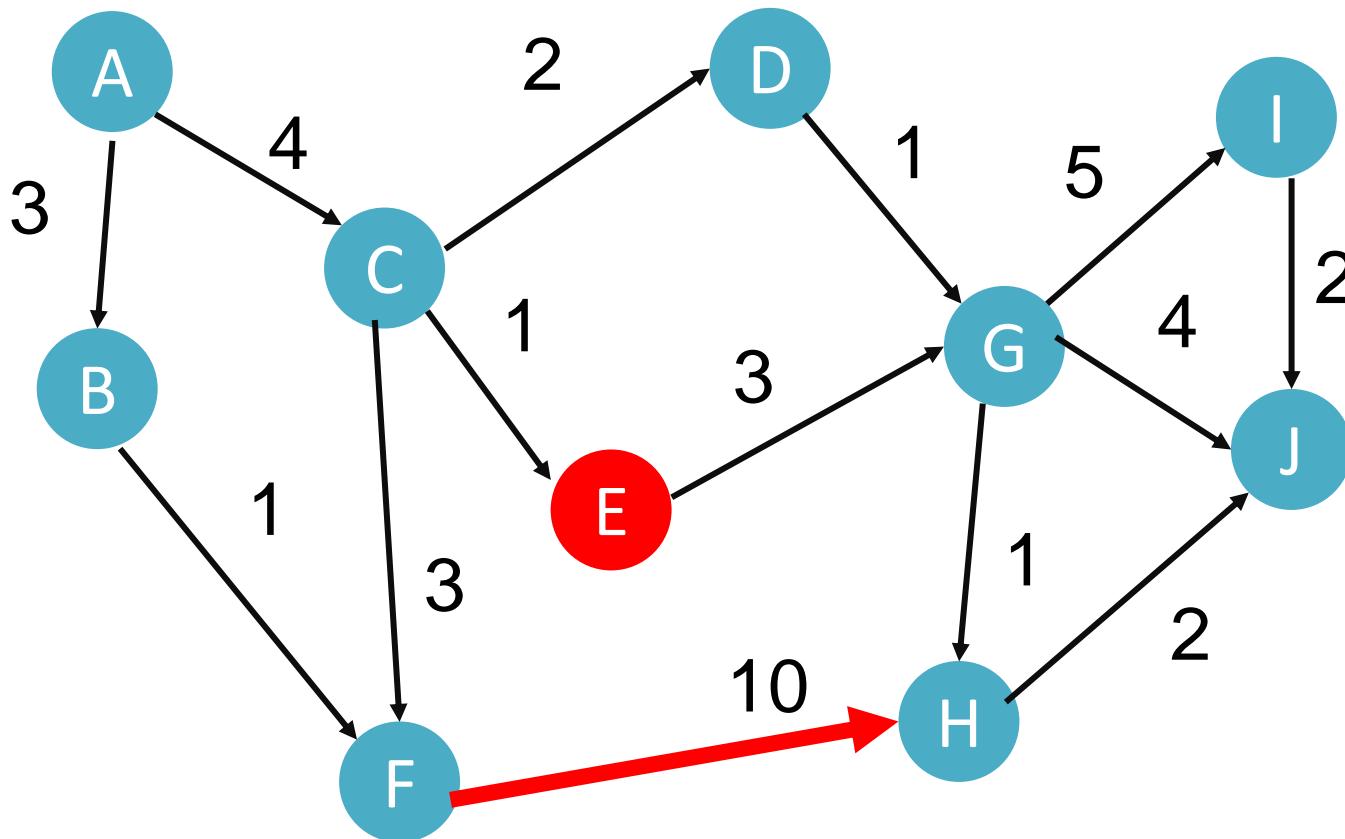
Tower of Hanoi

- กติกา 1. ให้อ้ายจาก 1-3  
2. ย้ายได้ครั้งละ 1 แผ่น  
3. ห้ามแผ่นใหญ่ทับแผ่นเล็ก

*ใครได้ จำนวนครั้งน้อยที่สุด ชนะ*



# SHORTEST PATH



## เส้นทาง

- ABFHJ = 16
- ACFHJ = 19
- ACEGIJ = 15
- ACEGJ
- ACEGHJ
- ACDGIJ
- ACDGJ
- ACDGHJ

**E** Server Down

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

## What is Event ?

- ยุค DOS / Console Application
- ยุค Desktop Application มีการทำงานแบบ GUI
  - *Event* Driven Programming / Visual Programming
  - OOP: Object Oriented Programming
- เว็บ Web Application
- Mobile Application

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

## การพัฒนาระบบ

- การพัฒนาระบบแบบดั้งเดิม/แบบโครงสร้าง/เชิงโครงสร้าง
  - เน้นการแก้ไขปัญหา โดยอัลกอริทึม เป็นเรื่องที่ค่อนข้างยาก
- การพัฒนาระบบเชิงวัตถุ
  - เน้นที่วัตถุ มองวัตถุเป็นหลัก

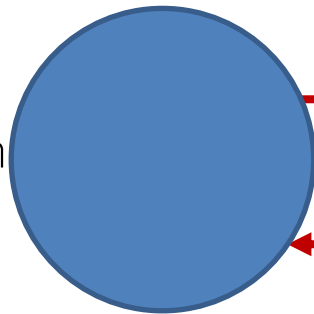
บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

Attributes

- ชื่อ
- นามสกุล
- รหัสนักศึกษา

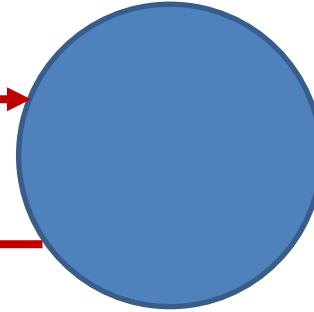
Functions

- ขอปรึกษา()
- เรียน()



นักศึกษา  
(Client)

ขอคำปรึกษา()



อาจารย์  
(Server)

ให้คำปรึกษา()



Attributes

- ชื่อ
- นามสกุล
- ตำแหน่ง

Functions

- ให้คำปรึกษา()

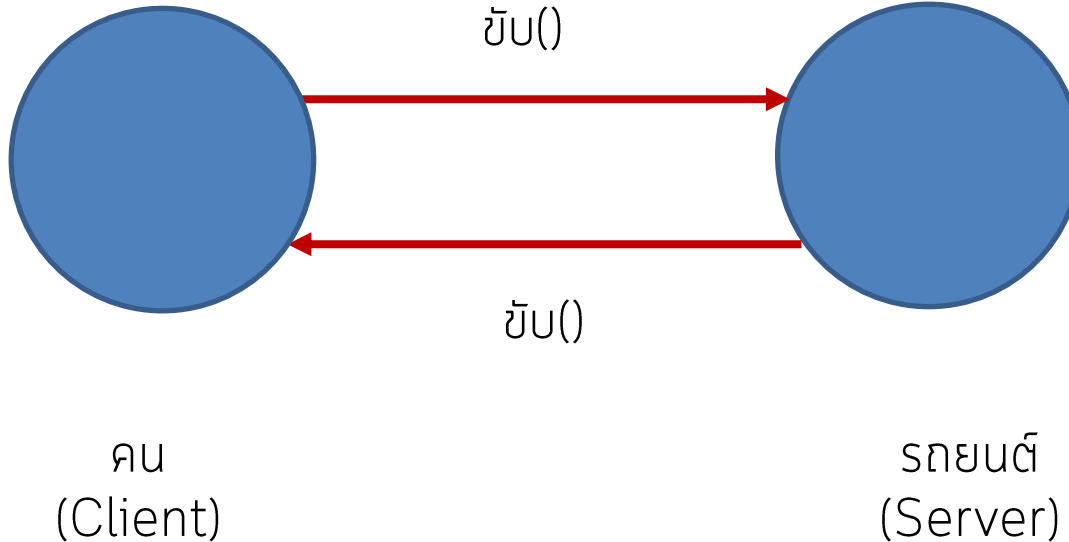
บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

Attributes

- ชื่อ
- นามสกุล

Functions

- ขับ()



Attributes

- ชื่อ
- นามสกุล
- ตำแหน่ง

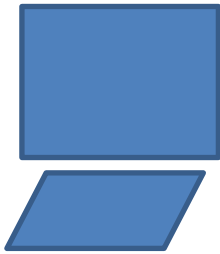
Functions

- วิ่ง()
- ทอยหลัง()
- เลี้ยว()

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

1-Tier

Word, accounting

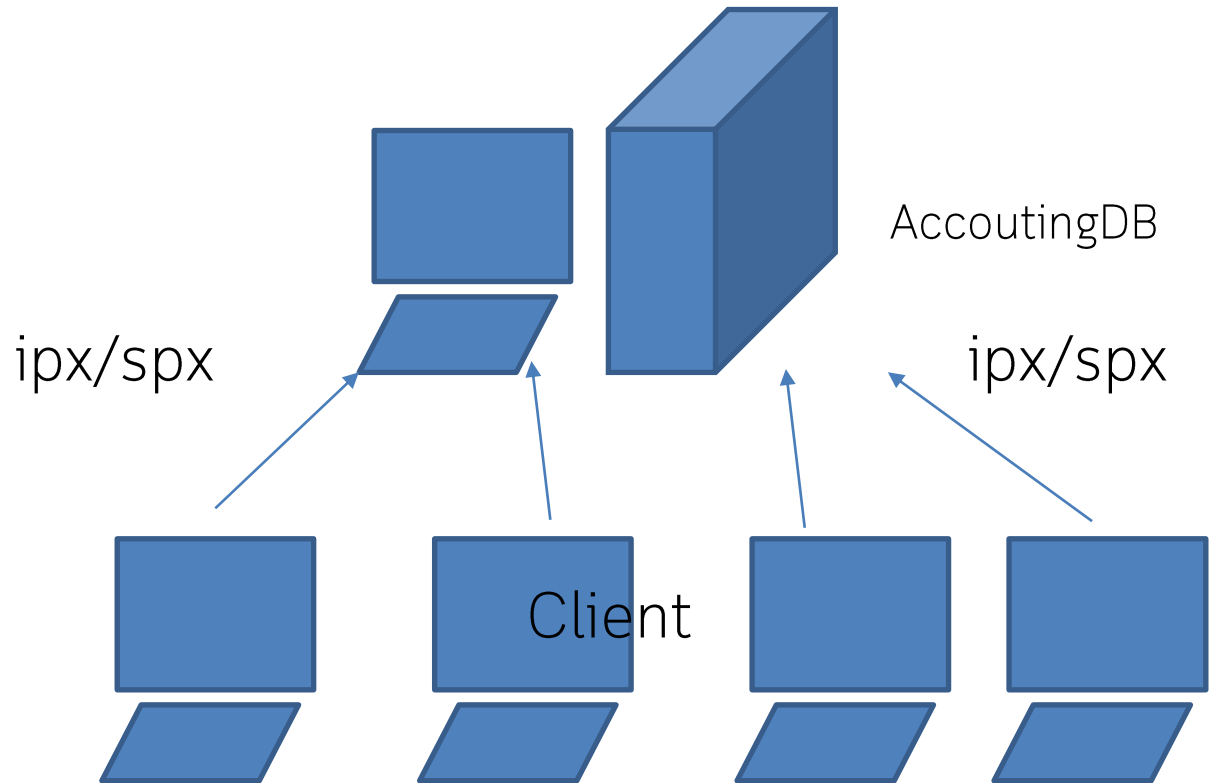


standalone

2-Tier

Client-Server

MIS , HRM , INVENTORY

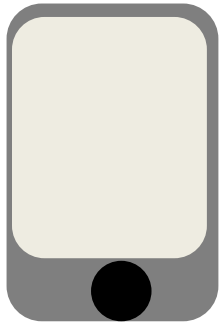


1-Tier

Word, Accouting



standalone



Smartphone

Tablet

Web-based

N-Tier

คลาวด์

Server

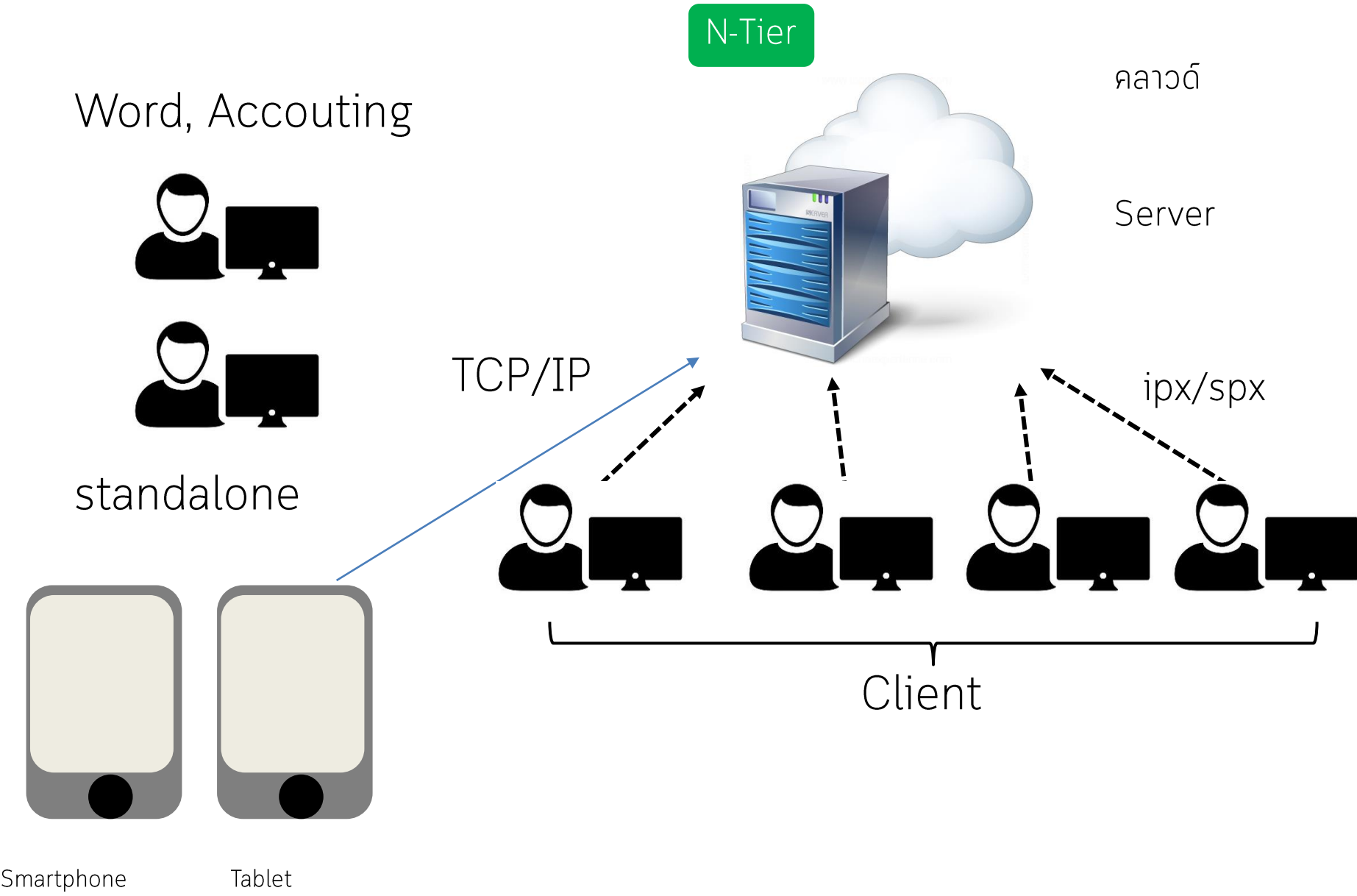


TCP/IP

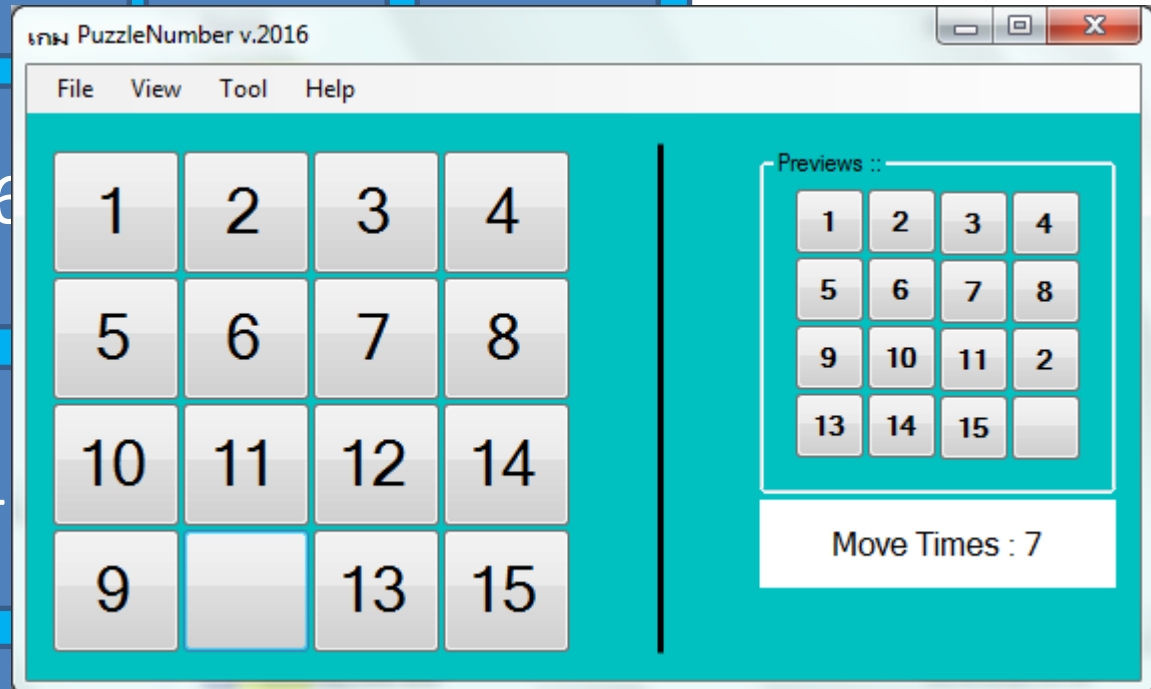
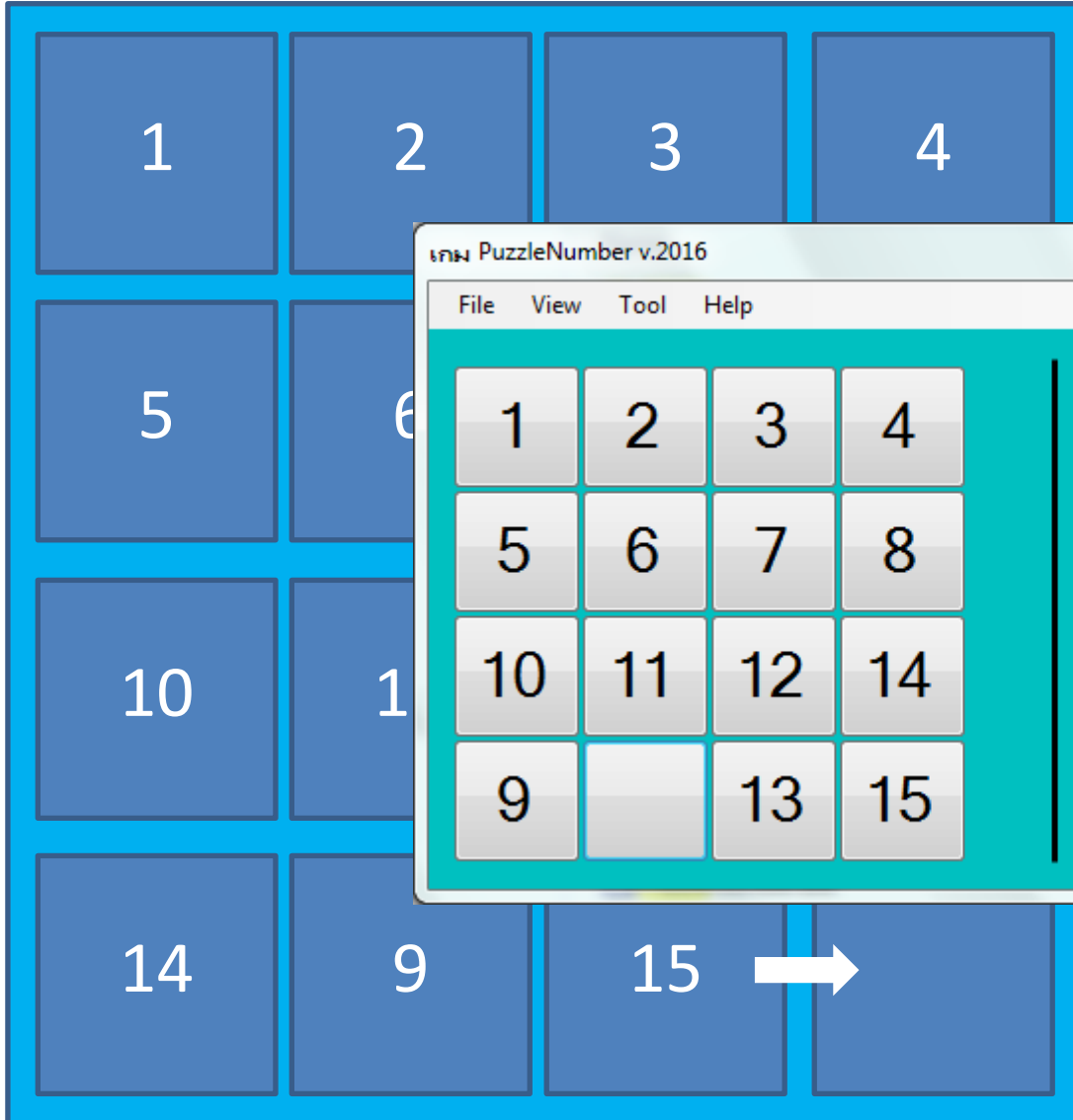
ipx/spx



Client

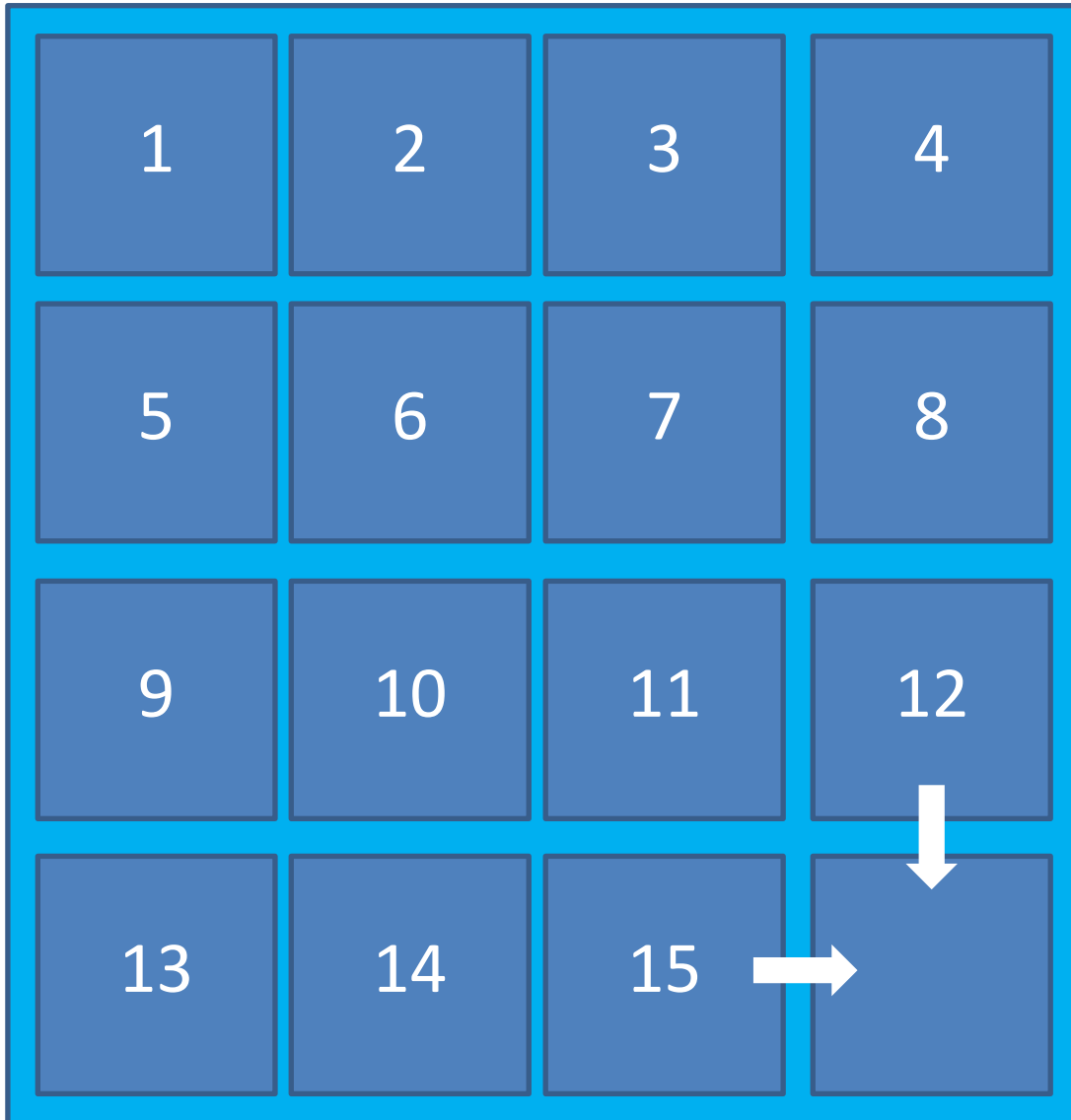


# Puzzle Number





# Puzzle Number



บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

## TicTacToe

O	X	O
	O	X
		O

8 9 5 6 1 7 5

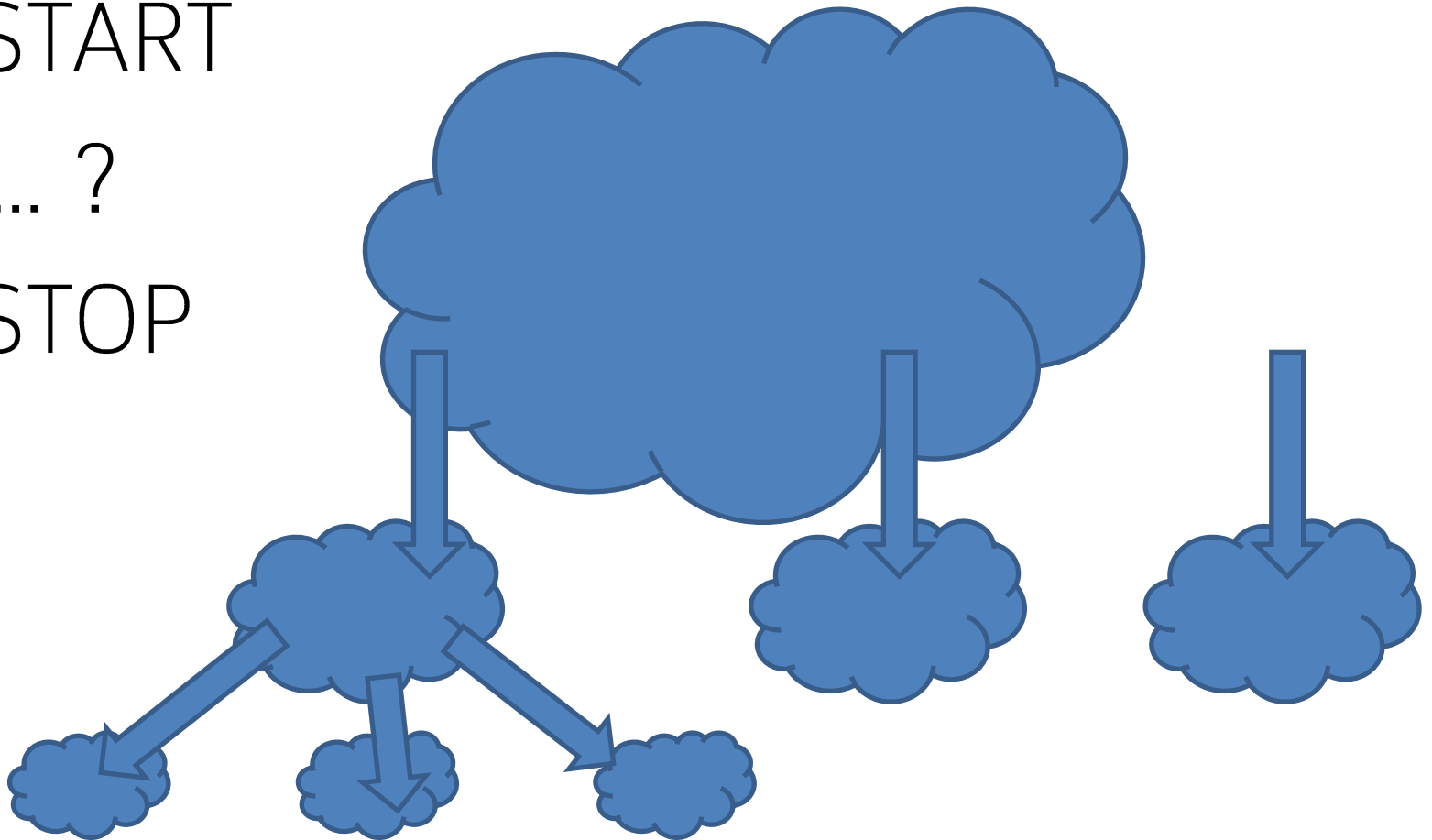
- PSEUDOCODE

**TOP- DOWN DESIGN**

–START

–.... ?

–STOP



8 9 5 6 1 7 5

- จงเขียน PSEUDO-CODE ของการเรียงลำดับ ตัวเลขที่กำหนดให้

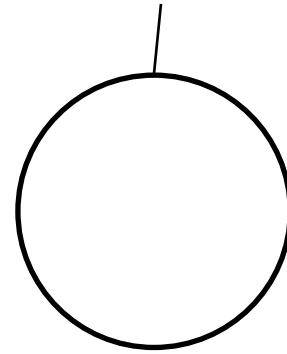
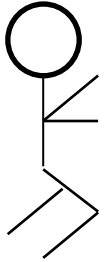
มองโจทย์ให้เป็นปัญหา แล้ว หาอัลกอริทึม มาแก้ไขปัญหา

1. เริ่มต้น
2. ...
3. สิ้นสุด

1. เริ่มต้น
2. ...
3. ...
4. ...
5. ...
6. สิ้นสุด

Pascal , C

uses graph;



$x1,y1$

$x2,y2$

```
plotline(x1,y1,x2,y2, 2);
```

# 1-100 มีเลขที่เป็น Prime Number อะไรบ้าง

- 1 กับ 2 ไม่นับว่าเป็น Prime number
- 3 5 7 ...97
- หาอย่างไร ?
- **อธิบาย หน้าห้อง**

หารเอาเศษ ใน VB

```
คำสั่ง if( x mod 2 = 0) then  
else  
end if
```

37

หารเอาเศษ ใน c, java , C#

```
คำสั่ง if( x % 2 == 0) {  
  
}
```

2..36

# Fibonacci Number

- **1 1 2 3 5 8 13 21 34 55 ....**

- **1- 100**

- **PSEUCODE**

- **FlowChart**

# ปัญหาของพัฒนาด้วยขั้นตอนวิธี (Algorithm)

อัลกอริทึม เป็นการแก้ไขปัญหาแบบมีขั้นตอน

คิดไม่ออก / ประสพการณ์

ไม่สามารถบอกออกมาเป็นข้อๆ ไม่ได้

ไม่ทราบขั้นตอนต่อไปต้องทำอะไร

จำโครงสร้างหลัก ๆ ไม่ได้

ถ้าเป็นผังงาน ก็ จำสัญลักษณ์ไม่ได้

ถ้าโจทย์ไหนที่ไม่เคยเจอ ก็ ยิ่งคิดไม่ออก



# Objects

- อะไรคือ วัตถุ ?

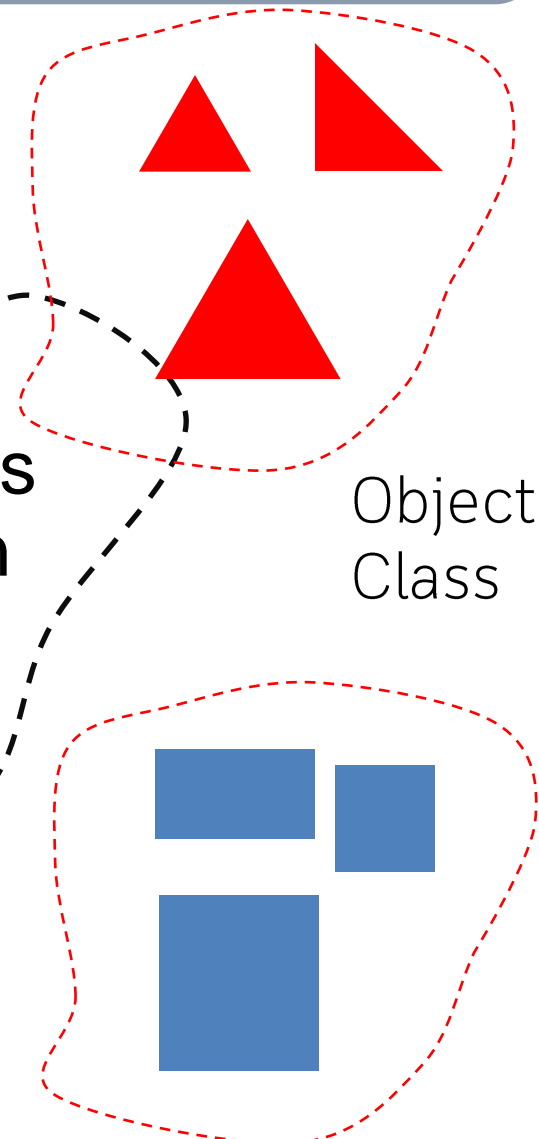
สิ่งที่สามารถบอกคุณลักษณะและ  
ความสามารถหรือฟังก์ก็ได้เรียกว่าวัตถุ

สิ่งที่ประกอบไปด้วย

- Attributes
- Function

- Attributes
- Function

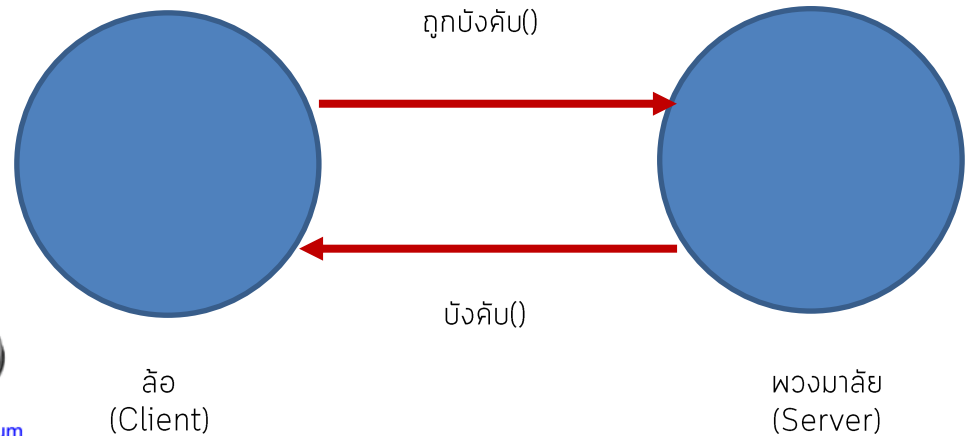
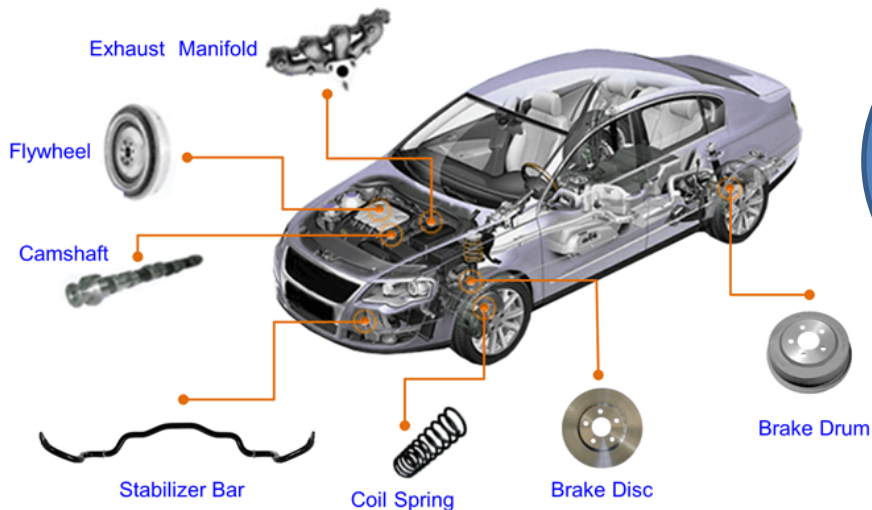
Object  
Class



บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

# วัตถุ รถยนต์

วัตถุต่าง ๆ ทำงานร่วมกัน



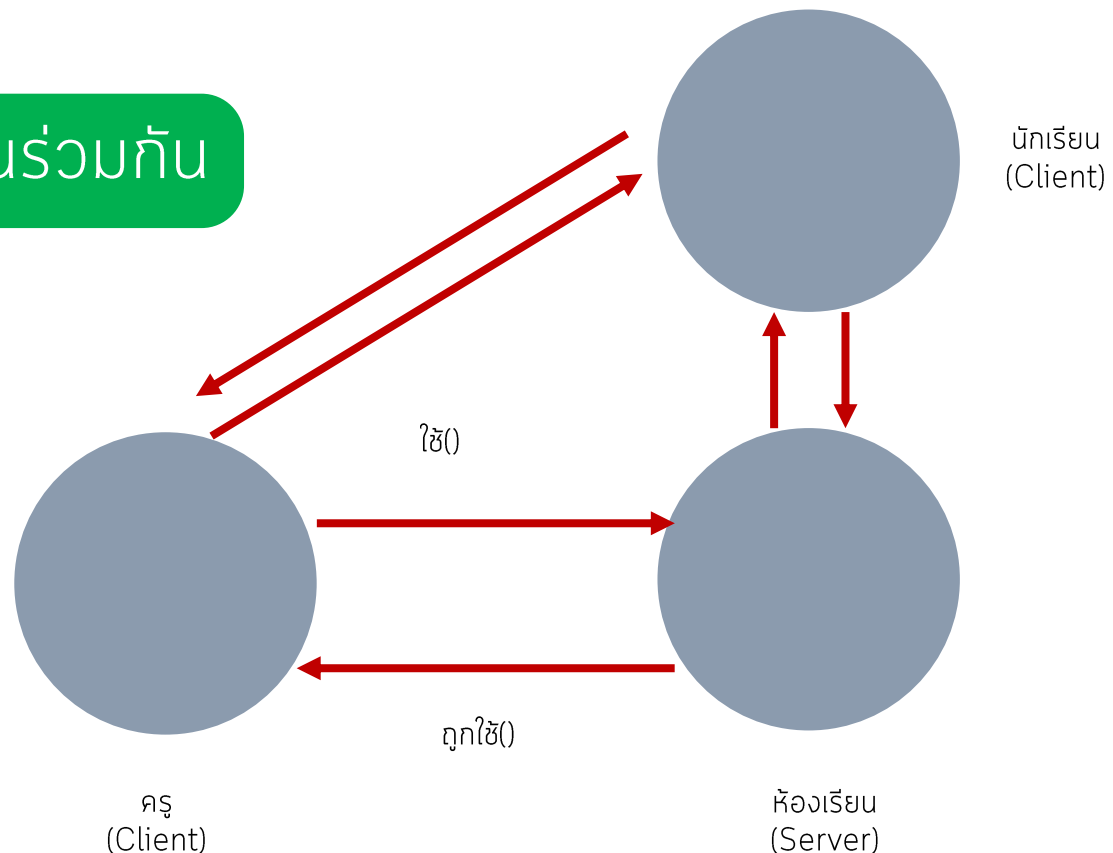
รถยนต์เกิดจากการทำงานร่วมกันของวัตถุต่าง ๆ เช่น ล้อ เบาะ พวงมาลัย กระจก เครื่องยนต์

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

# โปรแกรมเชิงวัตถุ

## วัตถุต่าง ๆ ทำงานร่วมกัน

- ครู
- นักเรียน
- ห้องเรียน
- รายวิชา



บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

วัตถุใน Visual Basic.net

TextBox

Label

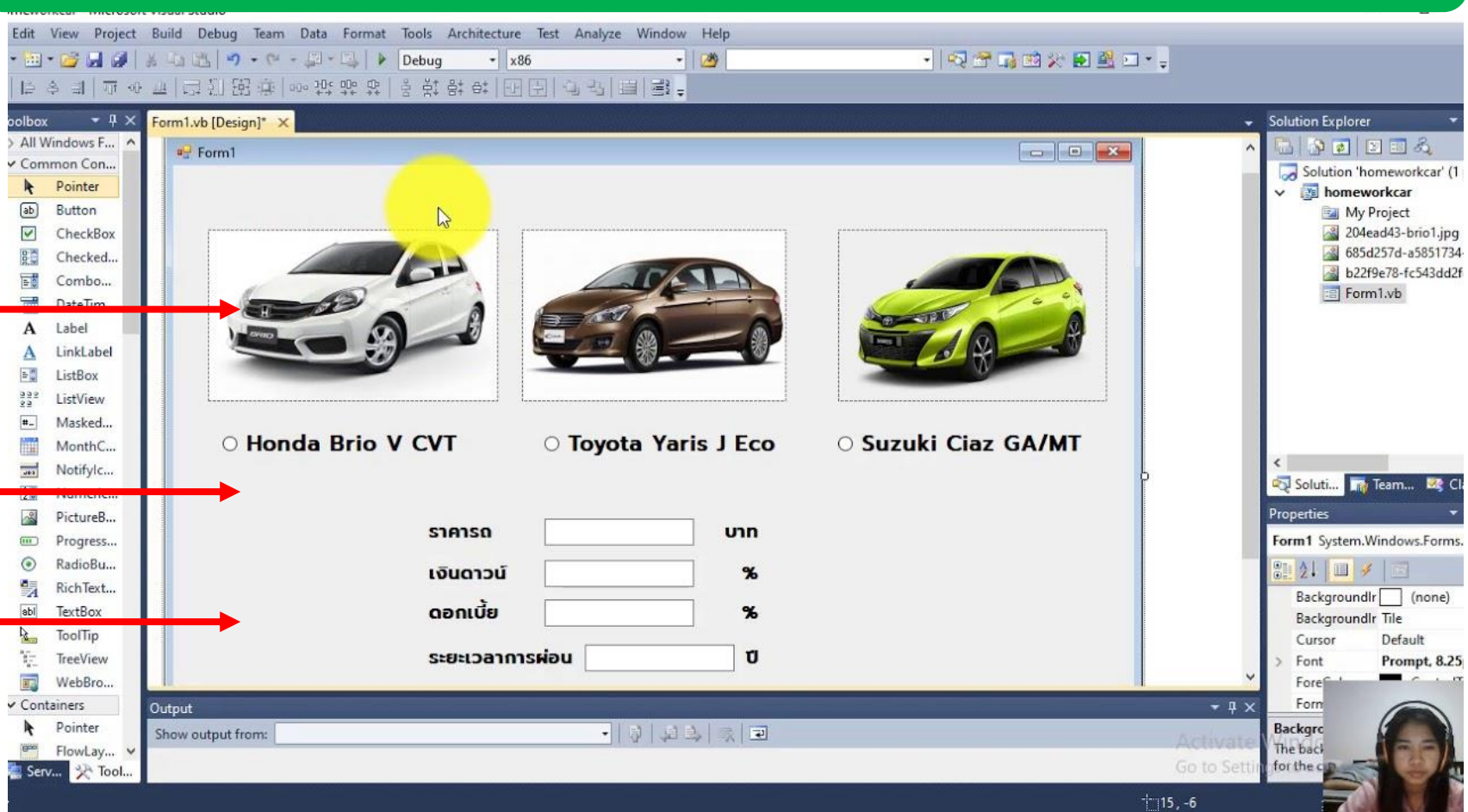
ListBox

Button

ใน Power Point สิ่งที่เป็นวัตถุได้ เมนู เครื่องมือ slide ฯลฯ  
ใน เมนู เอกสาร พิมพ์

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

วัตถุใน Visual Basic.net

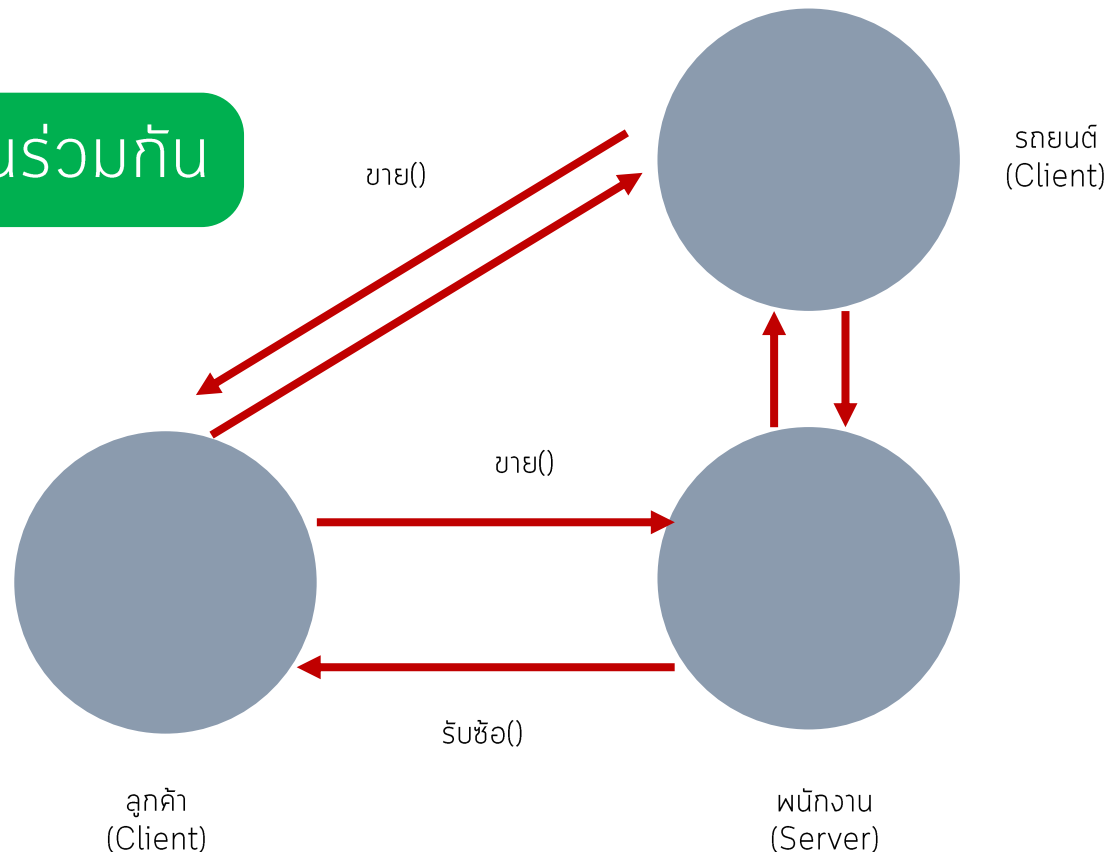


บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

ระบบซื้อ-ขายรถยนต์มือสอง

วัตถุต่าง ๆ ทำงานร่วมกัน

- ลูกค้า
- รถยนต์
- พนักงาน



# อะไรคือวัตถุ ในห้องเรียน

Interaction / Activity

Relationship

สอน Activity

อาจารย์ กับนักเรียน ...

Relationship

- อาจารย์
- นักเรียน
- เก้าอี้
- โต๊ะ
- โปรเจกต์เตอร์
- กระดานดำ
- หลอดไฟ



- Tangible Object วัตถุที่มองเห็น จับต้องได้ (รูปธรรม)
- Intangible Object เป็นวัตถุที่มองไม่เห็น หรือจับต้องไม่ได้ (นามธรรม)

OO  
OOP

# อาจารย์

- ชื่อ
- นามสกุล
- สี่นิ้ว
- น้ำหนัก
- ส่วนสูง
- เพศ
- สอน()
- เช็คชื่อ()
- ให้เกรด()

Attribute/

Vb: Properties/Data Type  
Java :Data Member

Function/behavior

Vb:Sub/function  
Java : Method



## บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม (Introduction to Object-Oriented and Abstractions)

- ความหมายของวัตถุ ตามหลักการ OO
- วัตถุ คือสิ่งที่เราสนใจ ซึ่งแบ่งออกเป็น 2 ประเภท
  - **Tangible Object** หมายถึงสิ่งที่ มองเห็น จับต้องได้ หรือรูปธรรม เช่น คน สัตว์ สิ่งของต่าง ๆ รถยนต์ สุขัข เก้าอี้ แมว ต้นไม้ รถประจำทาง เครื่องบิน คอมพิวเตอร์ พนักงาน ธนาคาร โรงแรม อะไหล่รถยนต์ เงิน (ธนบัตร) ฯลฯ
  - **Intangible Object** หมายถึง สิ่งที่ มองไม่เห็น จับต้องไม่ได้ หรือ สิ่งที่เป็นนามธรรม เช่น กฎหมาย กฎระเบียบ ข้อบังคับ เงิน(มูลค่า) เนื้อหาภาพยนตร์ ซอฟต์แวร์ เป็นต้น

# Tangible Object



# Intangible Object

- กฎเกณฑ์ หรือ กฎหมาย (ที่ไม่ใช่รูปเล่ม)
- ราคาสินค้า (ไม่ใช่ object แต่เป็น attribute ของสินค้า)
- อากาศ
- เวลา
- ความรู้ , ทักษะต่าง ๆ
- ภาพยนตร์ (ตัวเนื้อหาของเรื่อง)
- เงิน (มูลค่า)
- ซอฟต์แวร์ (โปรแกรม หรือชุดคำสั่ง)
- ฯลฯ

# องค์ประกอบ หรือ ส่วนประกอบ ของวัตถุ

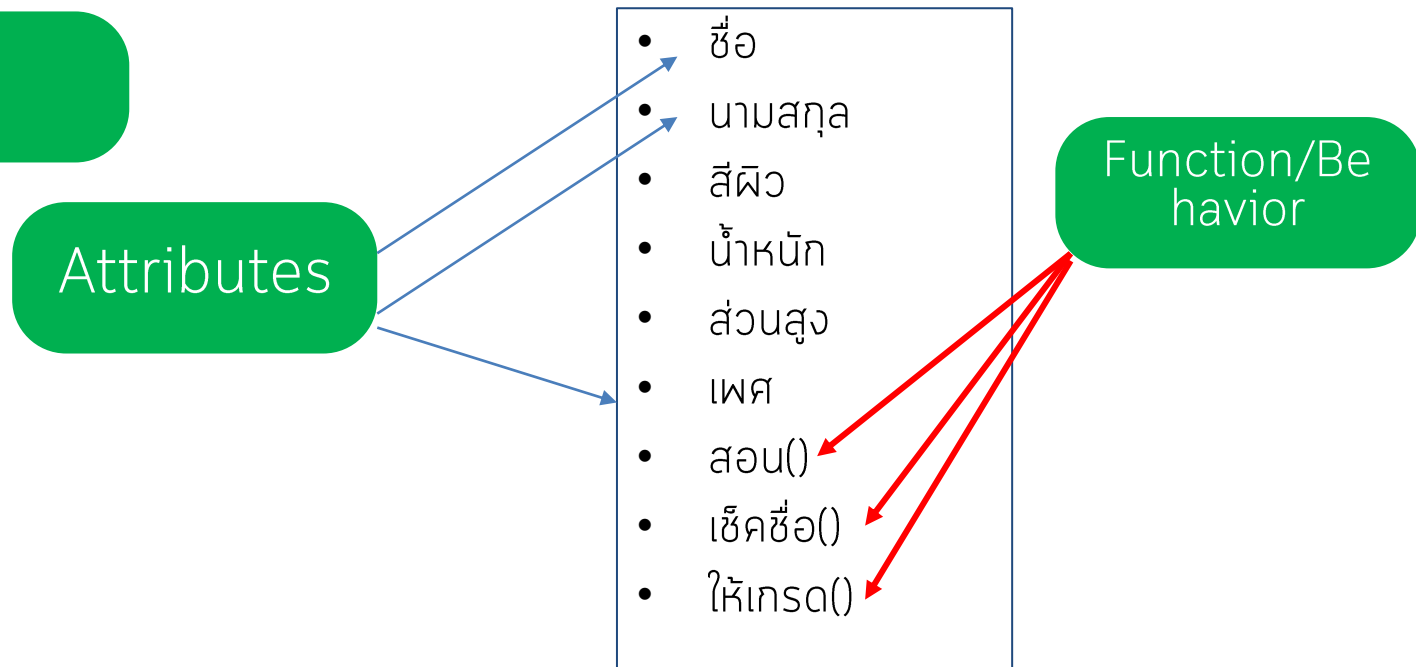
- วัตถุ จะประกอบไปด้วย 2 อย่าง
  - Attribute คุณลักษณะของวัตถุ ที่บ่งบอกถึงความเป็นวัตถุนั้น
  - Behavior หรือ Function พฤติกรรมหรือฟังก์ชันของวัตถุ

อาจารย์

Attributes

- ชื่อ
- นามสกุล
- สีผิว
- น้ำหนัก
- ส่วนสูง
- เพศ
- สอน()
- เช็คชื่อ()
- ให้เกรด()

Function/Behavior



# OO - โครงสร้างมือถือ

Real world



Computer world

- Attribute

- สี
- หน้าจอ
- รูปทรง
- .....
- ขนาด
- น้ำหนัก

- Function / Behavior

- โจร
- รับสาย
- ส่งข้อความ
- ถ่ายรูป



- Attribute

- สี
- หน้าจอ
- ยี่ห้อ
- ราคา
- รุ่น
- น้ำหนัก

- Function

- บอกชื่อ()
- บอกรุ่น()
- โจร
- ส่งข้อความ
- ถ่ายรูป

# OO - รถยนต์

Real world



Computer world

- Attribute

- สี
- รุ่น
- ยี่ห้อ
- ป้ายทะเบียน
- เบาะ
- พวงมาลัย

- Function / Behavior

- วิ่ง
- สตาร์ท
- หยุด
- เลี้ยว



- Attribute

- รุ่น
- ยี่ห้อ
- ราคา
- สี

- Function

- บอกชื่อ()
- บอกรุ่น()
- กำหนดราคา()
- กำหนดสี()
- กำหนดยี่ห้อ()

# Concept & Abstraction

- *Concept* ความคิดรวบยอด เกี่ยวกับสิ่ง ๆ นั้น สิ่งที่เราให้ความเห็น สรุปสาระสำคัญ ของสิ่งนั้น
- *Abstraction* กระบวนการที่เราให้นิยาม หรือความคิดรวบยอดเกี่ยวกับวัตถุในโลกความจริงเพื่อเข้าสู่โลกของคอมพิวเตอร์

Real World VS Computer World

# Real World



มุ่งเน้นลักษณะทางกายภาพมากกว่า

- ล้อ
- เบาะ
- ที่นั่ง
- พวงมาลัย

# Computer World



มุ่งเน้นข้อมูลมากกว่า

- ราคา
- รุ่น
- ยี่ห้อ
- สี



Real World

vs

Computer World

Real World

นก ..... ปีก ขา ขน → attribute  
..... บิน เดิน นอน ส่งเสียงร้อง → function

Computer World

นก ..... ชื่อ สายพันธ์ นน. ส่วนสูง → attribute  
..... บินได้ เดินได้ นอนได้ ส่งเสียงร้อง → function  
บอกชื่อ กำหนดชื่อ กำหนดสายพันธ์ → function

# การให้นิยาม(การให้ความคิดรวบยอดเกี่ยวกับ นก)กับ นก



นก ..... ชื่อ สายพันธ์ นน. ส่วนสูง → attribute  
..... บินได้ เดินได้ นอนได้ ส่งเสียงร้อง → function  
บอกชื่อ กำหนดชื่อ กำหนดสายพันธ์ → function

Real World

vs

Computer World

Real World

รถยนต์ ..... ล้อ ตัวถัง พวงมาลัย กระจก เบาะ → attribute  
..... สตาร์ทได้ หยุด เลี้ยว → function

Computer World

รถยนต์ ..... ยี่ห้อ รุ่น เลขทะเบียน สี → attribute  
..... บอกยี่ห้อ กำหนดยี่ห้อ บอกสี กำหนดเลขทะเบียน → function

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Oriented and Abstractions)

# เครื่องบิน

## Real World

เครื่องบิน .....ปีก ล้อ พวงมาลัย กระจก เบาะ ตัวเครื่อง ก่อ →  
attribute

.....สตาร์ท วิ่ง บินขึ้น ลงจอด → function

## Computer World

เครื่องบิน.....ยี่ห้อ รุ่น จำนวนที่นั่ง ราคาต่อที่นั่ง flight → attribute

..... บอกยี่ห้อ บอกรุ่น บอกราคาที่นั่ง → function

บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

concept (ความคิดรวบยอด) สิ่งนั้นคืออะไร

- 1. กลมก็ได้ ไม่กลมก็ได้
- 2. มี ก้าน .....
- 3. ก็นได้ ....
- 4. สีเหลือง ..
- 5. มีหนาม ...
- 6. ....

อมยิ้ม  
ไอติม  
ผลไม้  
ทุเรียน

ขุ่น

สีปะสด

# concept สิ่งนั้นคืออะไร

- มีหน้าจอ
- คีย์บอร์ด
- เมาส์
- ซีพียู
- ความสามารถในการประมวลผล ()
- เปิดได้ ()
- เล่นเน็ตได้ ()
- ปิดได้()

คอมพิวเตอร์



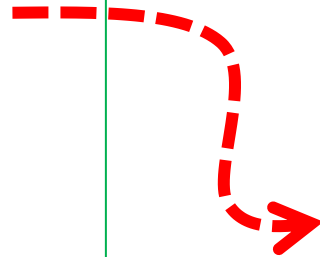
คอมพิวเตอร์สีแดง

คอมพิวเตอร์ Acer

# Real World & Computer World

## ■ รถยนต์

- ล้อ
- หลังคา
- ตัวถัง
- เคลื่อนที่ไปข้างหน้า
- เลี้ยวได้
- หยุดได้
- วิ่งได้



## ■ รถยนต์

- ยี่ห้อ
- รุ่น
- ราคา
- สี
- ป้ายทะเบียน
- + เปลี่ยนชื่อ()
- + เปลี่ยนยี่ห้อได้()
- + กำหนดราคาได้()
- + กำหนดรุ่นได้()

# Real World & Computer World

## • โทรศัพท์

- .....ยี่ห้อ
- .....รุ่น
- .....ราคา
- .....สี
- เลขตัวเครื่อง
- กล้อง
- หน้าจอ
- ปุ่ม
- สายชาร์จ
- ช่องเสียบ
- Sim
- ลำโพง
- + โทรได้()
- + รับสาย()
- + ส่งข้อความ()
- + ถ่ายรูปได้()

## • โทรศัพท์

- ราคา
- รุ่น
- ยี่ห้อ
- สี
- เบอร์โทร
- หน่วยความจำ
- กำหนดราคาได้()
- กำหนดสี()
- บอกราคาได้()
- บอกสีได้()
- กำหนดรุ่นได้()
- บอกรุ่น()
- ~~เปลี่ยนจอได้()~~
- ~~เปลี่ยนผู้ใช้ได้()~~
- เปลี่ยนรุ่น()
- เปลี่ยนซิมได้()
- เปลี่ยนแบตเตอรี่()



บทที่ 2 : การนำไปสู่หลักการเชิงวัตถุแนวคิดเชิงนามธรรม  
(Introduction to Object-Orientation and Abstractions)

## Concept : ความคิดรวบยอด

- มี กระจก .....
- ล้อ .....
- มีปีก ....
- หน้าต่าง
- บินได้()
- โดยสารได้()

....หน้าต่าง

เครื่องบิน

# วัตถุทางคอมพิวเตอร์ กับ วัตถุในโลกความจริง

- วัตถุ ในความเป็นจริง บางครั้งอาจไม่นิยามเข้าไปในคอมพิวเตอร์ เพราะใช้ประโยชน์ไม่ได้ หรือไม่สมารถทำได้
- **ล้อ** คือ วัตถุในโลกความเป็นจริง คือสิ่งที่ใช้หมุน ใช้กับรถยนต์ หรือยานพาหนะ
- **ยาง ดอกยาง ก้าน แม็ก**
  - ในทางคอมพิวเตอร์ wheel **ชื่อ รุ่น ยี่ห้อ ราคา จำนวนล้อ**
  - **พฤติกรรม หรือ ฟังก์ชัน เปลี่ยนชื่อ ได้ เปลี่ยนรุ่น ใส่ราคา กำหนดราคา**

คลาส กับ วัตถุ  
Class & Object

# What is class ?

อาจารย์

นักเรียน  
นักศึกษา

Class คือกลุ่มของวัตถุ

ภัคจิรา มีชื่อ เบอร์โทร สอนได้

ซูไรดา มีชื่อ เบอร์โทร สอนได้

มินตรา มีชื่อ เบอร์โทร สอนได้

อรนุช มีชื่อ มีรหัส ลงทะเบียนได้

ศศิพส มีชื่อ มีรหัส ลงทะเบียนได้

Group of the objects  
Type of the objects

# What is class ?

Class คือกลุ่มของวัตถุ

ถ้าเราเจอก้อนกล้วย เรา จะจัดให้ ก้อนกล้วย อยู่ในกลุ่ม นี้ หรือไม่

ก้อนกล้วย

ต้นสน

ต้นสัก

คะน้า

พืช

ผักกาด

ผักบุ้ง

แมว

ถ้าเราไปเจอ สิ่งที่มีใบ สีเขียว อยู่บนพื้นดิน มีราก

Group of the objects  
Type of the objects

# วัตถุ และคลาส

- TextBox1
- TextBox2
- TextBox3

**Instance / Object** หรือวัตถุ คือตัวแทนของคลาส  
ดังนั้นการชี้เฉพาะเจาะจงลงไป คือ วัตถุ หากพูดโดยรวมจะ  
หมายถึงคลาส

- TextBox

- TextBox1
- TextBox2
- TextBox3
- TxtUN
- TxtPWD

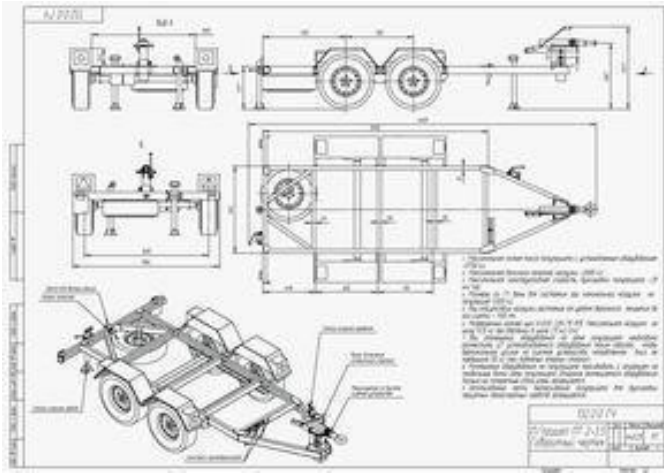
Button

OK

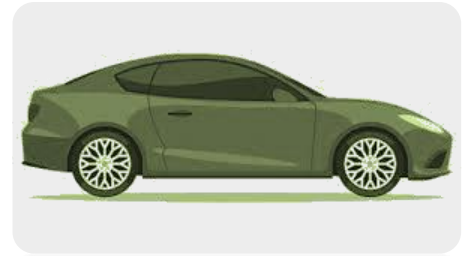
Cancel

Close

# โรงงาน Factory → Class



instantiation



คลาสเป็นแม่แบบในการสร้างวัตถุ



สีรถยนต์ = สีเขียว

สีรถยนต์ = สีแดง

ถ้าให้นิยามผิดให้ความคิดรวบยอดผิด จะทำให้เกิดปัญหา

แมว

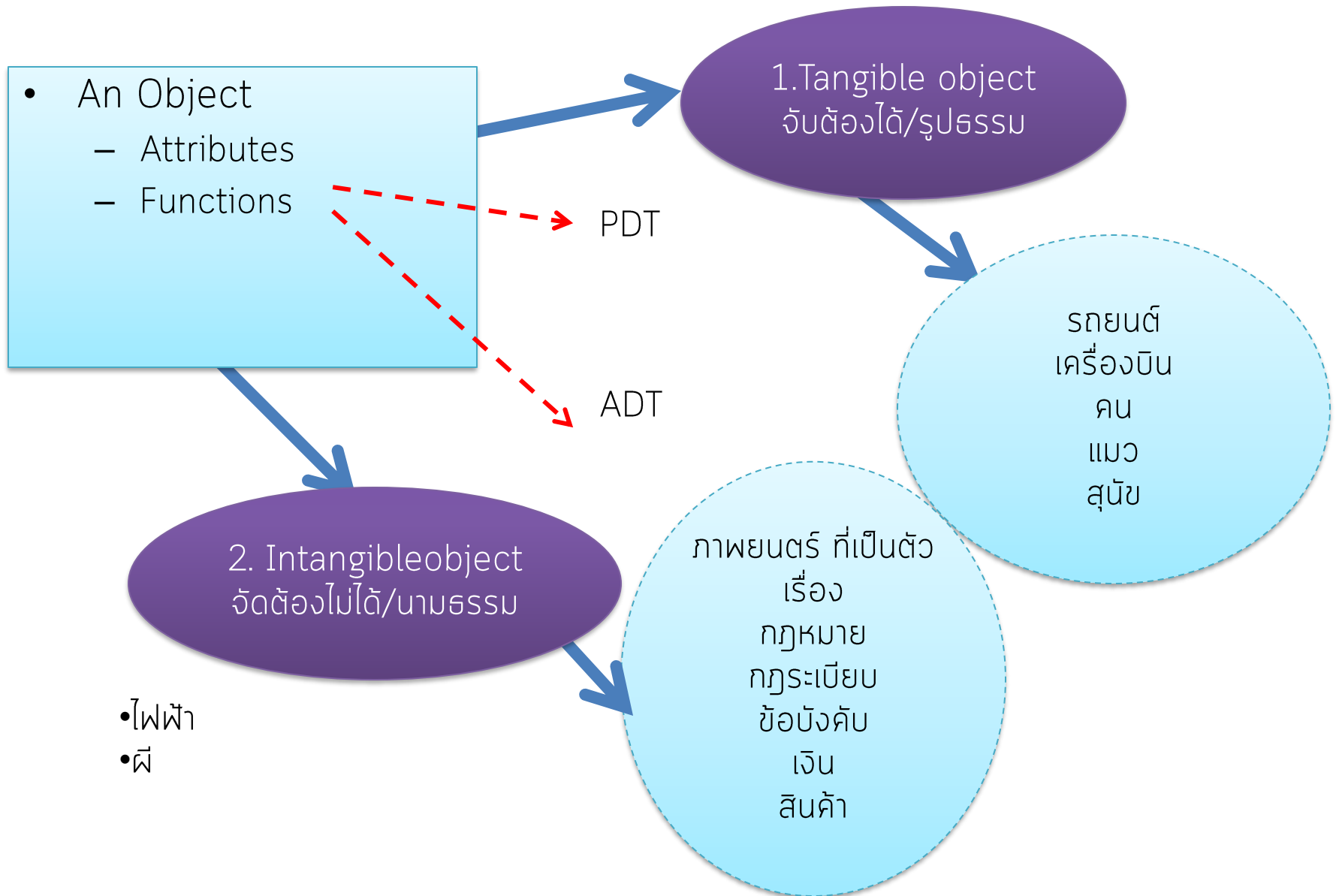
- มีใบ
- มีราก
- มีกิ่ง

- + อยู่บนพื้นดิน()
- + ดูดซึบสารอาหารได้()
- + สังเคราะห์แสงได้()

พืช ?



# OO: object orientation



# สิ่งที่ นศ. ต้องทำความเข้าใจก่อน

- อะไรคือคลาส/class
- อะไรคือ วัตถุ/object

ตู้เย็น  
ตู้เย็น LG  
Acer  
โน้ตบุ๊ก  
โน้ตบุ๊ก Acer aspire 550  
iPhone 14s  
iPhone 14s สีแดงของสม  
โทรศัพท์ ซัมซุง S23 สีแดง  
Samsung  
โต๊ะ  
โต๊ะอาจารย์ ...มีหลายตัว class  
โต๊ะอาจารย์ห้อง 845 ...มีตัวใน  
ห้องเรียน object  
เก้าอี้

สิ่งใดเรียกว่า คลาส และสิ่งใดเรียกว่า วัตถุ

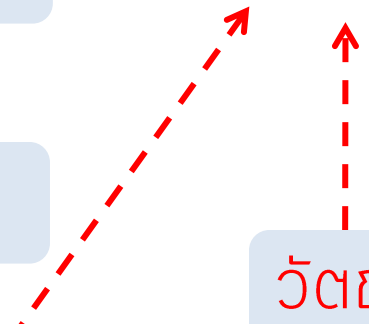
ตู้เย็น เป็นคลาสหรือ วัตถุ ?

ตู้เย็น เป็นคลาส

คลาส

iPhone 6s

วัตถุ



ตู้เย็น เป็นคลาส

## จะทำให้ตู้เย็นเป็นวัตถุ

- ระบุ สี ระบุความเป็นเฉพาะ เช่น สี ยี่ห้อ ความเป็นเจ้าของ
- ตู้เย็นยี่ห้อ Mitsubishi สีแดง ของ อ.ดร. นัฐพงศ์ ส่งเนียม

- รถยนต์ เป็นคลาส
- **รถยนต์ ยี่ห้อ BMW สีแดง ป้ายทะเบียน 4ก5443**

- นักศึกษา เป็นคลาส
- นักศึกษา สมชาย ใจดี

- อาจารย์ เป็นคลาส
- อาจารย์ นัฐพงศ์ เป็นวัตถุจากคลาสอาจารย์

# จงบอกว่สิ่งต่อไปนี้เป็นคลาสหรือเป็นวัตถุ เพราะอะไร

ตู้เย็น ..... อลงกรณ์ .....คลาส ..... ไม่ได้เจาะจงลงไป ...

LG ..... ประยុท ..... วัตถุ ยี่ห้อ การระบุยี่ห้อลงไปคือการเจาะจง LG เป็นวัตถุ  
หนึ่งหรือ ยี่ห้อหนึ่งของคลาสเครื่องใช้ไฟฟ้า

Acer .....ภควัต ..... Acer เป็นวัตถุหนึ่งหรือ ยี่ห้อหนึ่งของคลาสเครื่องไ้ตบู้ค

ไ้ตบู้ค .....ภคมัย เป็นคลาส .....

iPhone 8s .....สุภาพร

S7 .....สายฟ้า

Samsung .....ภมรธิดา

ไ้ตะ .....วัชราวุฒิ

เก้าอี้ .....วงรดา....

คลาส

แก้วน้ําเยติ

แก้วน้ําเยติ

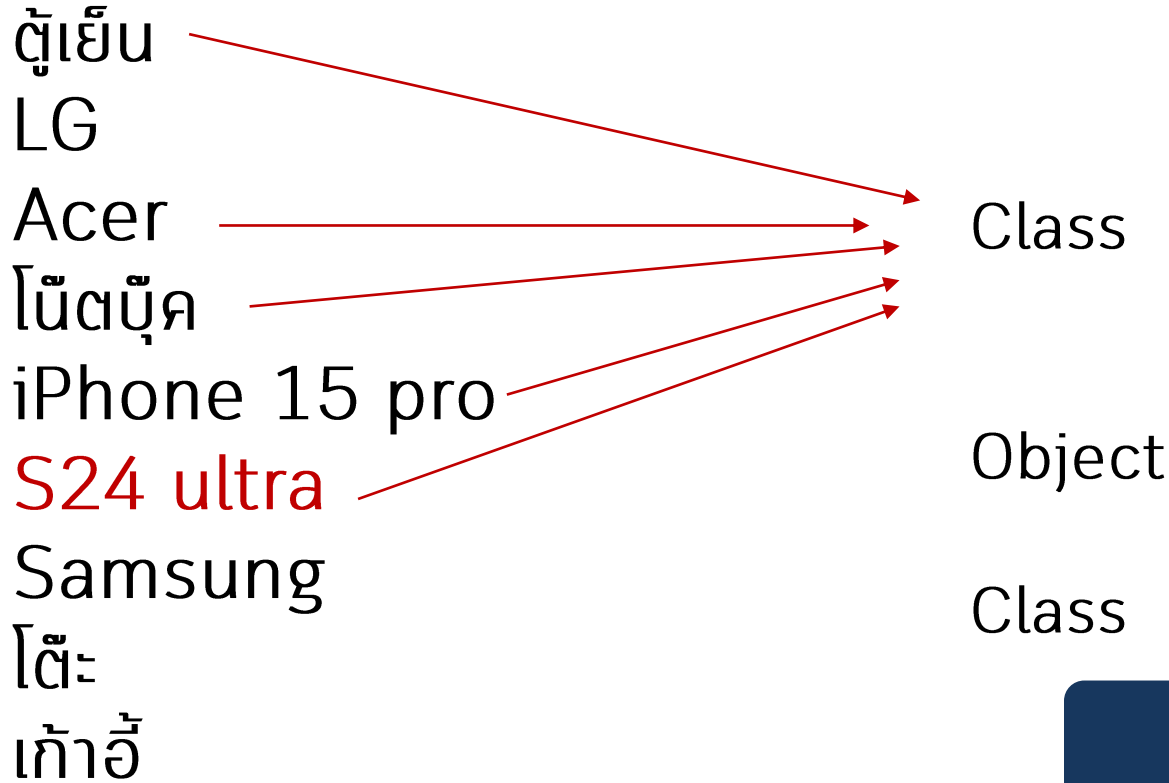
???

เก้าอี้

คลาส



ตัวอย่างที่ 2.1 จงบอกว่สิ่งใดต่อไปนี้ เรียกว่า คลาส และสิ่งใดเรียกว่า วัตถุ



Object

iPhone 15 pro สีดำ ของนายวรินทร์

S24 ultra ของ อ. นัฐ สีเงิน

## \*\*\* การทำให้คลาส เฉพาะเจาะจงลงไปเพื่อทำเป็นวัตถุ

- เพิ่มความเป็นเจ้าของ โต๊ะ >> โต๊ะของนายสมชาย
- สี >> โทรศัพท์ >> โทรศัพท์เครื่องสีแดง
- ยี่ห้อ >> โทรศัพท์เครื่องสีแดงยี่ห้อ iPhone
- หรือ รุ่น >> โทรศัพท์เครื่องสีแดงยี่ห้อ iPhone รุ่น iPhone13 ProMax
- หรือความสัมพันธ์ เพิ่มเติม
- เช่น computer คือ คลาส ถ้าบอกว่า คอมพิวเตอร์ สีแดง ยี่ห้อ Acer ของนายดำ ก็เป็น วัตถุ

## แบบฝึกหัด 2.1

### จงหาคลาส และ วัตถุ จาก domain นี้

สมชายเป็นนักศึกษาสาขาวิชาวิทยาการคอมพิวเตอร์ ภาคปกติ  
รุ่น 65 นั่งเขียนโปรแกรมภาษา C# ด้วยเครื่องโน้ตบุ๊ก Acer  
สีแดง รุ่น Aspire 5301 และขณะเดียวกันสมหญิงนั่งคุย  
โทรศัพท์กับเพื่อนด้วยเครื่อง iPhone15 proMax

จงหาคลาส และ วัตถุ จากข้อความด้านบนแล้วเขียนลงในกระดาษ แล้วถ่ายรูปส่ง  
มาทางไลน์ โดยใส่ google drive ด้านล่างนี้ ใส่ชื่อและรหัส 13 หลัก (ก่อนวัน  
จันทร์ หัก)

## แบบฝึกหัด 2.2

### จงทำให้คลาสต่อไปนี้เป็นวัตถุ

---

- รถยนต์
- คอมพิวเตอร์
- ตู้เย็น
- คน
- แมว
- เครื่องปรับอากาศ
- ธนาคาร



## แบบฝึกหัด 2.3

### จงระบุว่าสิ่งใดเป็นคลาสสิ่งใดเป็นวัตถุ

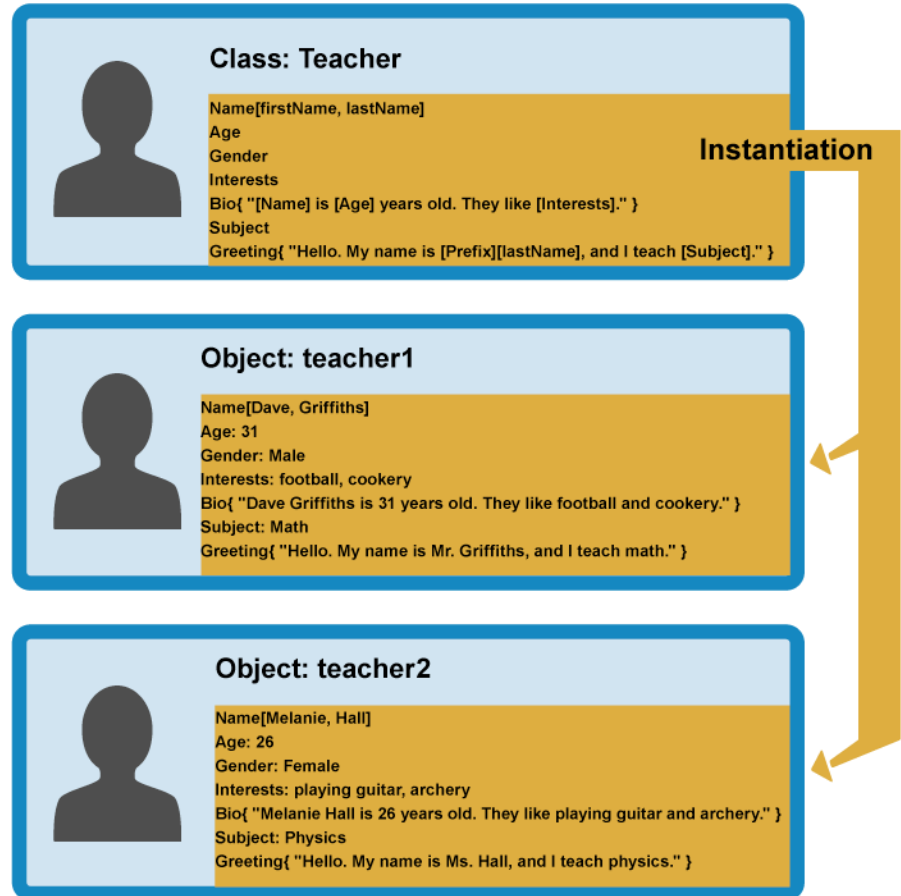
1. Cat
2. Dog
3. Rain
4. Planet
5. Earth
6. Pencil
7. Tom and jerry
8. Snoopy
9. Mickey mouse
10. Money

# สรุปคำศัพท์

- Problem domain
- Class
- Object
- Concept
- Computer world
- Real world
- Abstraction

# Object Instantiation

คลาสเปรียบเสมือนแม่แบบ ที่ใช้  
ในการสร้างวัตถุขึ้นมา

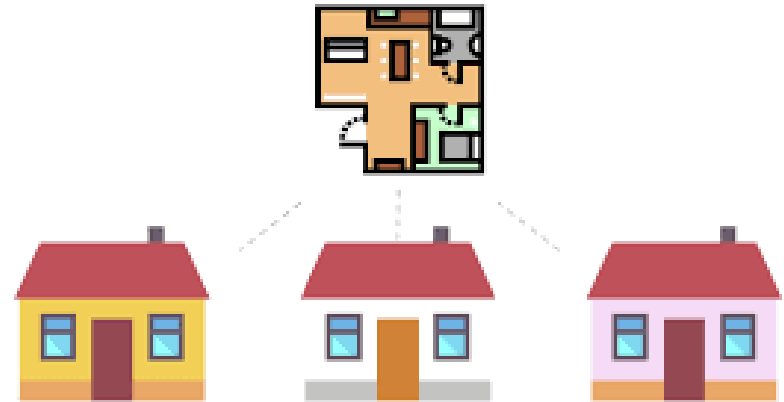
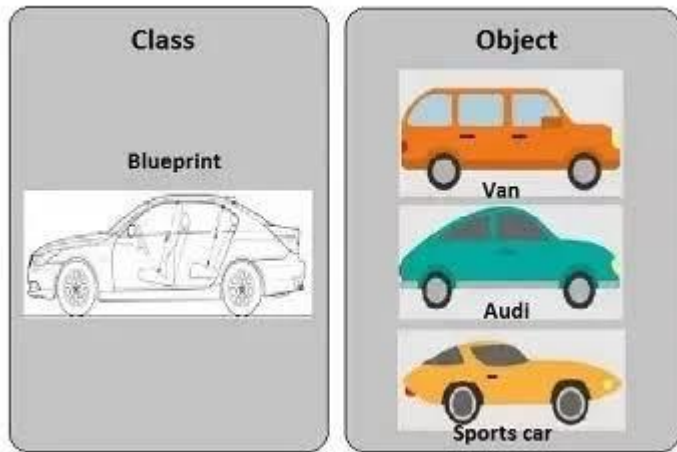


# Object Instantiation

บ้านแต่ละหลังมาจาก  
คลาสเดียวกัน



# Class and Object

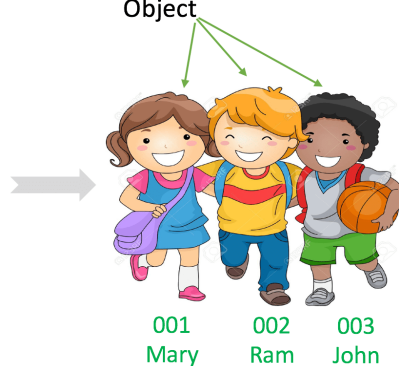


Class Student

```
Int RollNo;  
String Name;
```

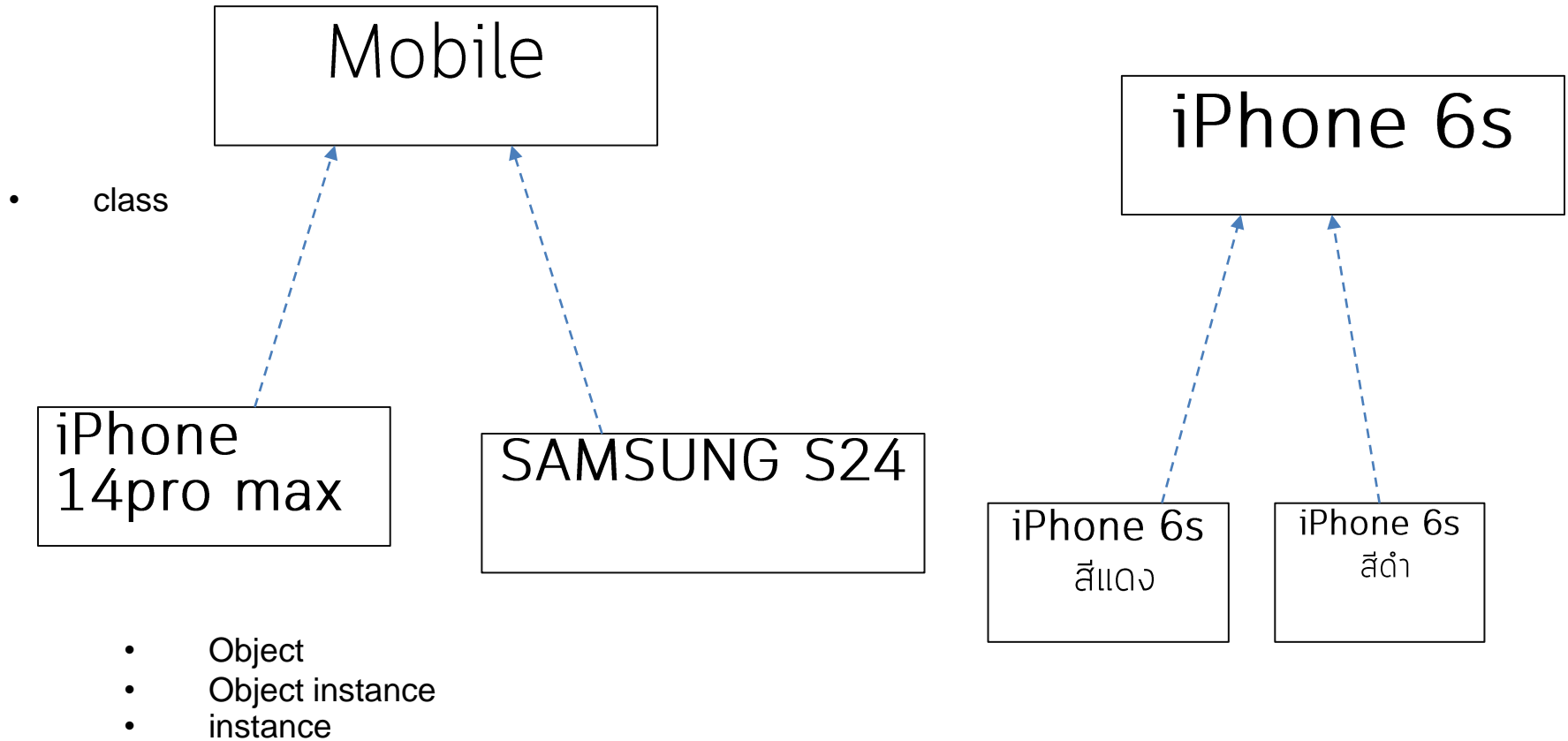
```
SetRoll();  
DispRoll();  
SetName();  
DispName();
```

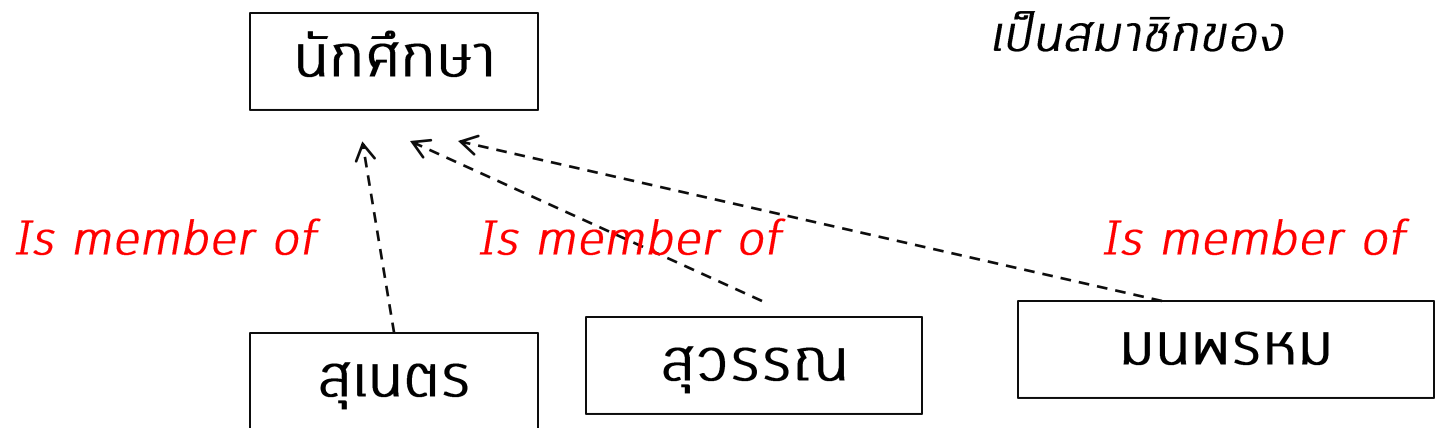
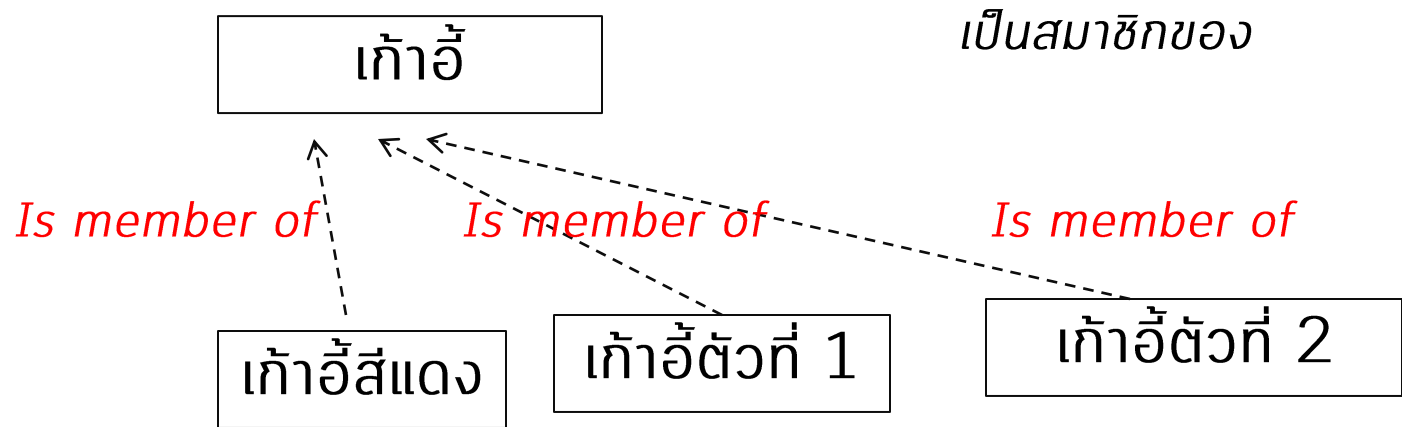
Object

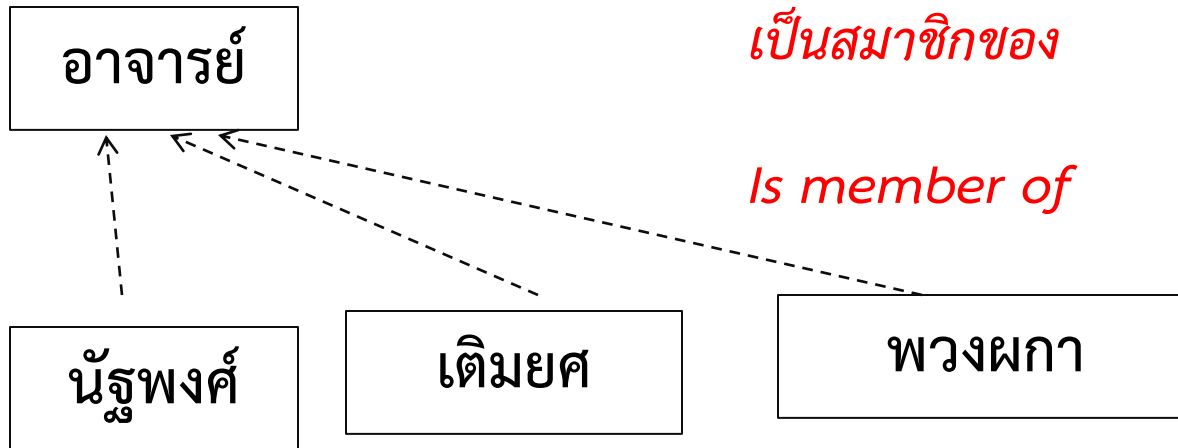


**วัตถุจะมีความเป็น  
identity หรือ unique**

# วัตถุและคลาส







## Attributes

ชื่อ  
ตำแหน่ง  
เบอร์โทร  
อีเมล

## Functions

สอน  
เช็คชื่อ  
ออกข้อสอบ  
ตรวจข้อสอบ  
ตัดเกรด



# Type / ชนิด

- 1 87 9 →

จำนวนเต็ม integer

- 

- 73.5 72.8 1.9 1.005

→

ทศนิยม floating-point

- A Z ก ข ..ฮ →

ตัวอักขระ char

**PDT : Primitive Data Type**

# Attributes

Type จะมี 2 ประเภท

## 1. PDT : *primitive data type*

int , byte , char , string , Boolean  
ตัวอักษร ตัวเลข ข้อความ ตรรกะ

## 2. ADT : Abstract data type

ชนิดที่เป็นวัตถุ

คน ชื่อ นามสกุล ที่ อยู่ >> PDT

แขน ขา หัว ลำตัว >> วัตถุ >> ADT

# คอมพิวเตอร์

- ยี่ห้อ PDT
- ราคา PDT
- เมาส์ ADT
- จอภาพ ADT



attribute



function

# Attributes : คุณลักษณะของวัตถุ

คือ สิ่งที่ วัตถุ มี หรือ แสดงให้เห็น

รถยนต์

- ล้อ
- เครื่องยนต์
- กระจก
- สี
- น้ำหนัก
- พวงมาลัย
- เกียร์

ADT

ADT

คน ไม่ใช่ attribute  
ของรถยนต์

น้ำมัน ไม่จัดเป็น  
attribute ของรถยนต์  
เพราะน้ำมัน คือวัตถุ อีก  
อันหนึ่งที่รถยนต์มี  
ปฏิสัมพันธ์  
(Interaction) ด้วย

# Problem Domain

- พัฒน คือ อะไร ?

Attributes ลักษณะของตัวพัฒนา มีเยอะ ก็ต้องเลือกเอาเฉพาะที่เราสนใจ เรียกว่าเป็น **Problem Domain** (เราไม่สนใจทุกสิ่งพัฒนาได้) แล้วอะไร คือสิ่งที่สนใจของพัฒนาดังนั้นถ้าพูดถึงพัฒนา คนอื่นๆ ต้องเข้าใจ

ถ้า อาจารย์ พูดถึง สิ่งมีมอเตอร์ มีใบพัด ใช้ไฟฟ้า มีปุ่มกดปรับความแรงได้ เรียกสิ่งนี้ว่าพัฒนา

 Hatari ใบพัด มีมอเตอร์ มีปุ่มกด ใช้ไฟฟ้า ปรับระดับความแรง และหมุนได้

# Problem Domain

- น็อต มีขา มีฐาน มีสายไฟ คือ Attributes ของพัดลมใช้หรือไม่

ถ้า อาจารย์ พูดถึง สิ่งที่มีมอเตอร์ มีใบพัด ใช้ไฟฟ้า มีปุ่มกด  
ปรับความแรงได้ เรียกสิ่งนี้ว่าพัดลม

 Hatari ใบพัด มีมอเตอร์ มีปุ่มกด ใช้ไฟฟ้า ปรับระดับความแรง และหมุนได้

## 2.1 Object Orientation คืออะไร

- เราสามารถพบเห็น สิ่งของต่าง ๆ มากมายในชีวิตประจำวัน ไม่ว่าจะเป็นสิ่งที่มองเห็นได้ (**Tangible**) และสิ่งที่มองไม่เห็น (**Intangible**)
  - ต.ย. ของ Tangible (รูปธรรม) ได้แก่ >> คอมพิวเตอร์ , ทีวี , โทรศัพท์มือถือ , รถยนต์ , คน , สัตว์ ฯลฯ
  - ต.ย. ของ Intangible (นามธรรม) ได้แก่ >> กฎเกณฑ์ หรือ กฎหมาย (ที่ไม่ใช่รูปเล่ม) , ราคาสินค้า , อากาศ , เวลา , ความรู้ , ทัศนคติต่าง ๆ ภาพยนตร์ (ตัวเนื้อหาของเรื่อง) ฯลฯ



# Tangible

# InTangible

เงิน

1. ตู้กด

2. รongเท้า

3. คางคก

1. ฝตู้กด

2. เชือกrongเท้า

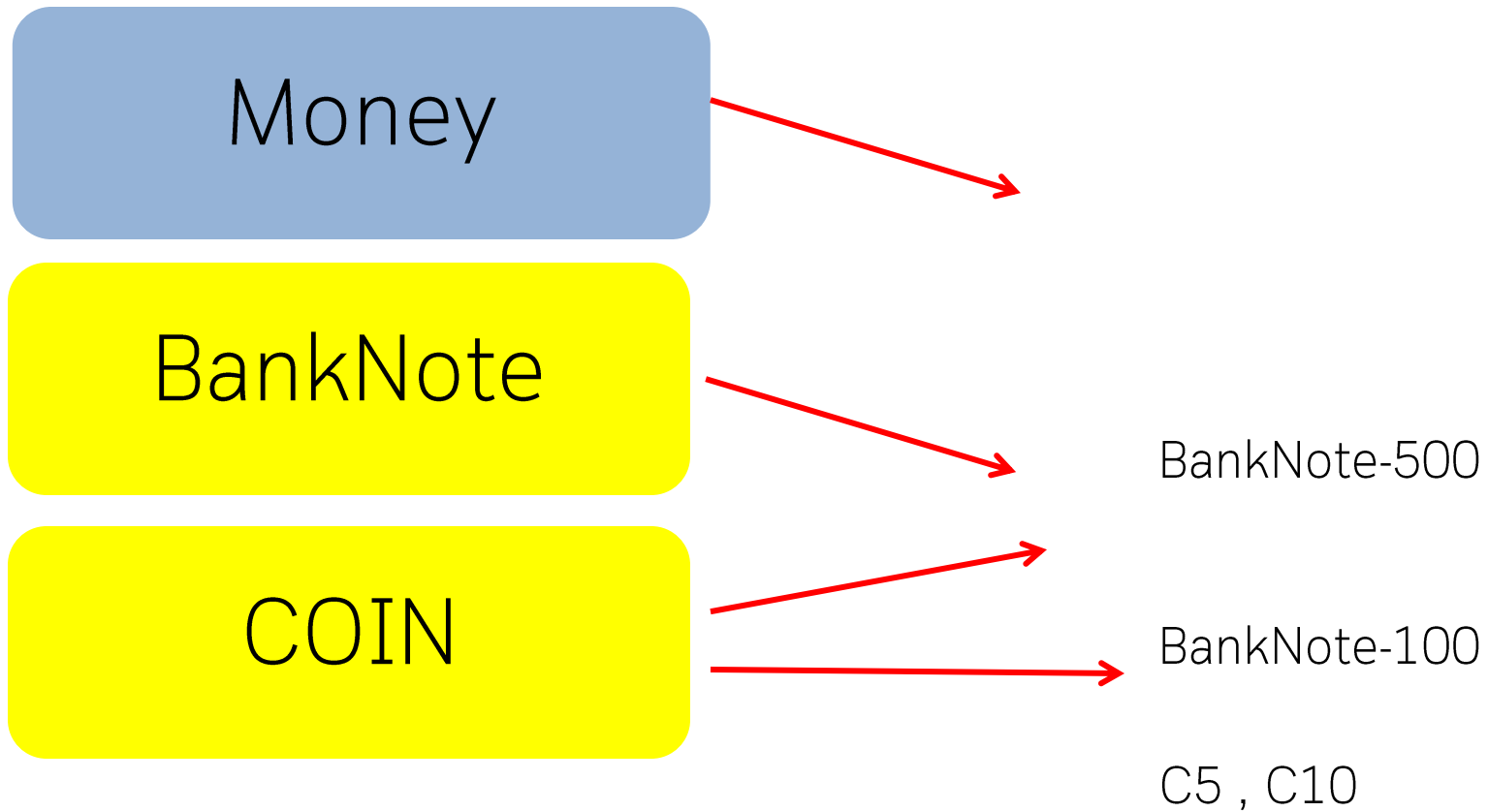
3. อารมณ

Attributes

Function







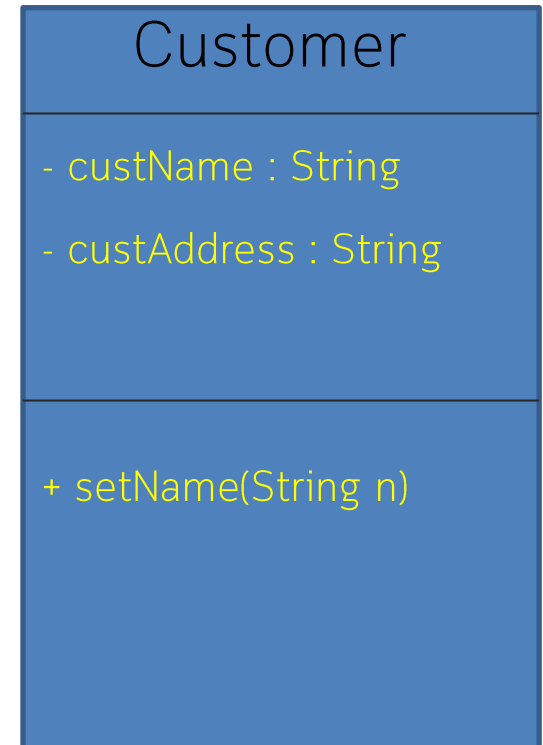
Please decription these information in the following word are ..What is Class , object , Tangible , Intangible , Attribute , Function ?

# รทยนต์

- อธิบาย Encapsulation ของรทยนต์
  - 1. คำอธิบาย คำว่า Encapsulation คืออะไร
  - **Encapsulation** คือการห่อหุ้ม หมายถึงการนำเอาลักษณะและความสามารถของวัตถุ รวมกัน ดังนั้นรทยนต์ก็มีความสามารถในการห่อหุ้ม คือการเอาชื่อ ฟิลด์ เครื่องยนต์ ที่นั่ง พวงมาลัย รวมกับความสามารถในการเคลื่อนที่ เลี้ยว หยุด
  - 2. Encapsulation ของรทยนต์ คืออะไร
- อธิบาย Information Hiding ของรทยนต์
  - Information Hiding คือ การซ่อนรายละเอียด
  - หมายถึง การที่ซ่อนรายละเอียดบางอย่างจากผู้ใช้งาน โดยที่ผู้ใช้งานไม่จำเป็นต้องรู้ ภายในเครื่องยนต์ ประกอบไปด้วยอะไรบ้าง ทำงานอย่างไร แต่ผู้ใช้งานรู้วิธีรทยนต์ ก็พอ ถูกออกแบบมาเพื่อให้ผู้ใช้งานรู้แค่บางอย่าง ที่จำเป็นเท่านั้น เพื่อความปลอดภัย

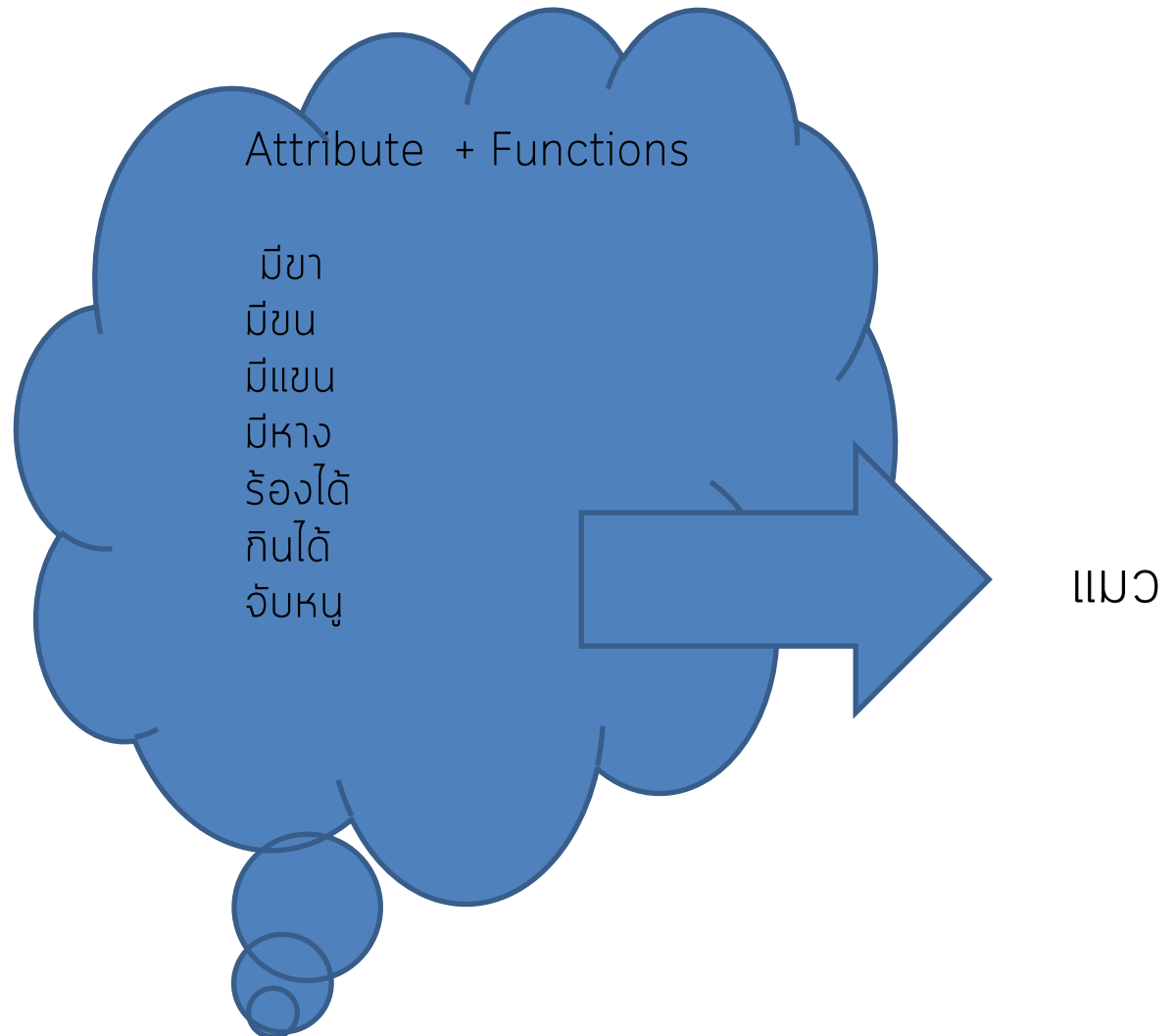
# คลาส customer

```
class Customer {  
    String custName;  
    String custAddress;  
  
    public void setName(String n) {  
        custName = n;  
    }  
}
```

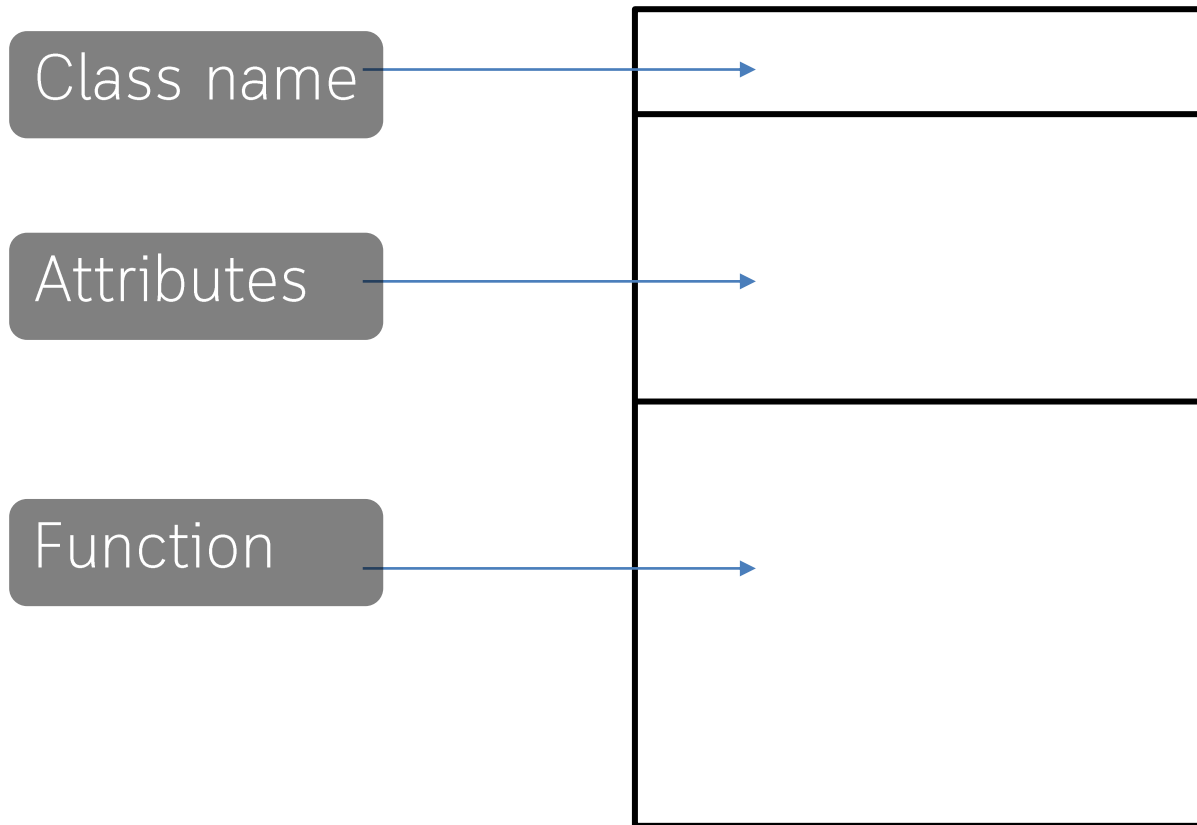


# Encapsulation **vs** Information hiding

- รถยนต์
- ตู้เย็น
- ทีวี
- โทรทัศน์
- รถไฟ
- เครื่องบิน
- แมว
- สุนัข



# UML: Class Diagram



รถไฟ

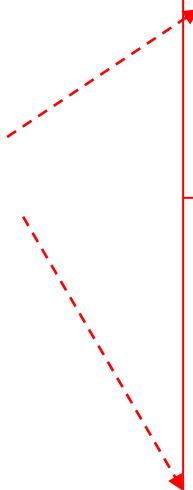
Class name

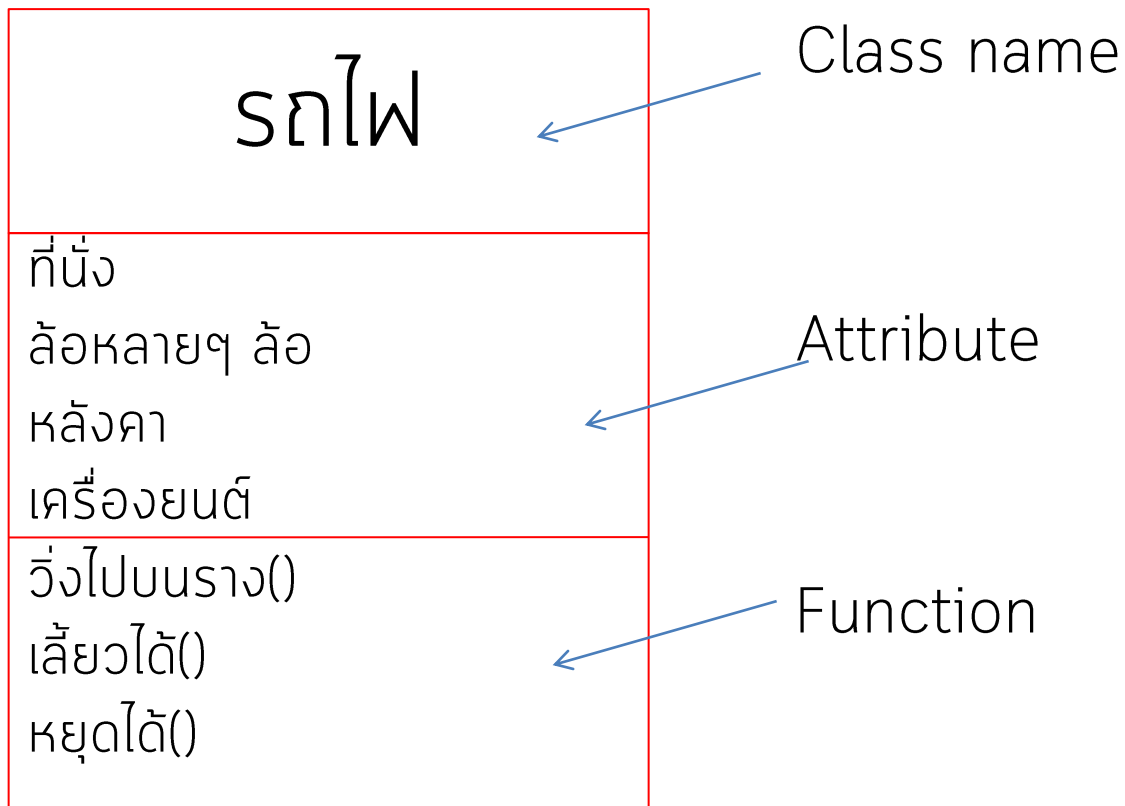
- ที่นั่ง  
ล้อหลายๆ ล้อ  
หลังคา  
เครื่องยนต์

Attribute

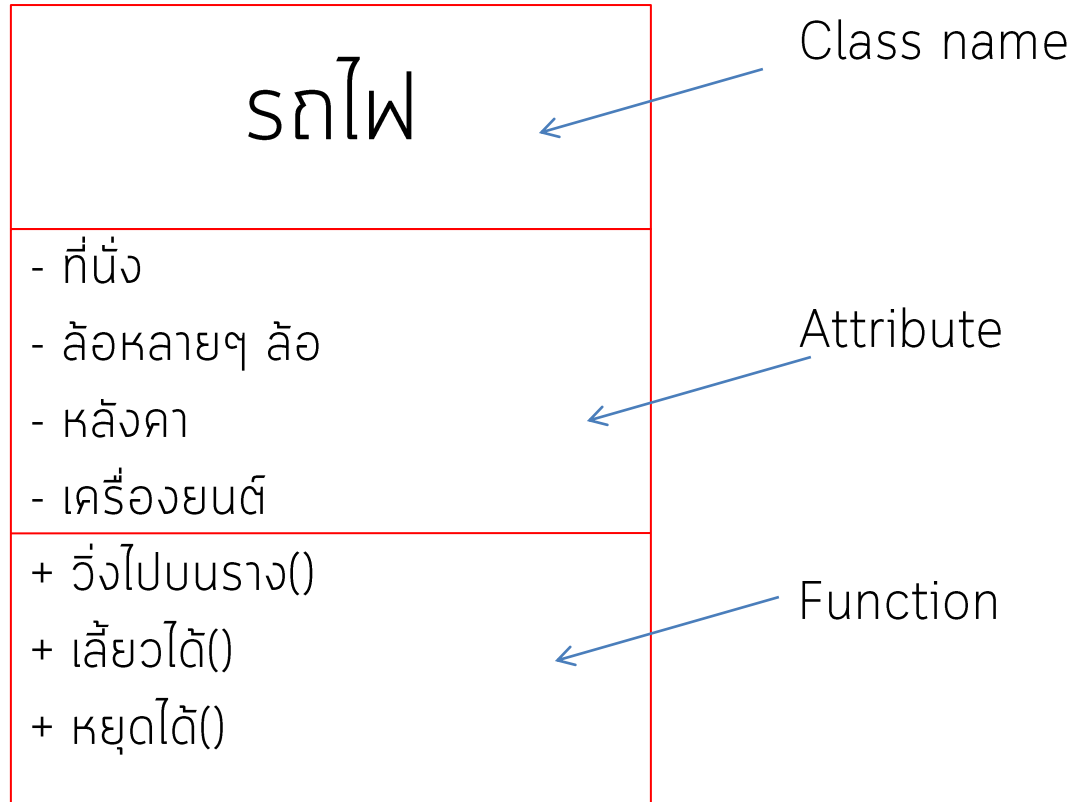
+ วิ่งไปบนราง()  
+ เลี้ยวได้()  
+ หยุดได้()  
+ กำหนดที่นั่งได้()  
+ สตาร์ทได้()

Function





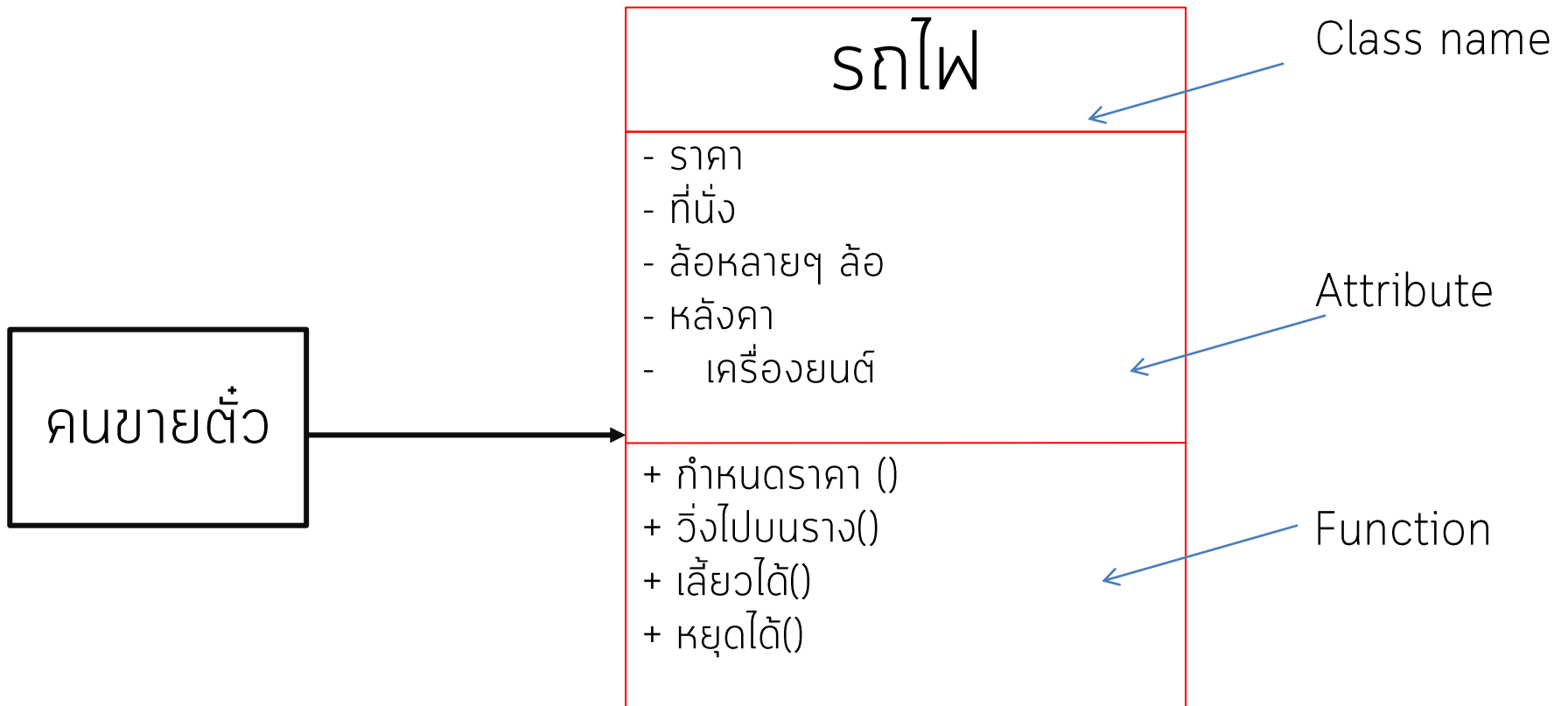
จงอธิบาย information hiding ของ รถไฟ



## Visibility/Accessibility

จง อธิบาย information hiding ของ รถไฟ



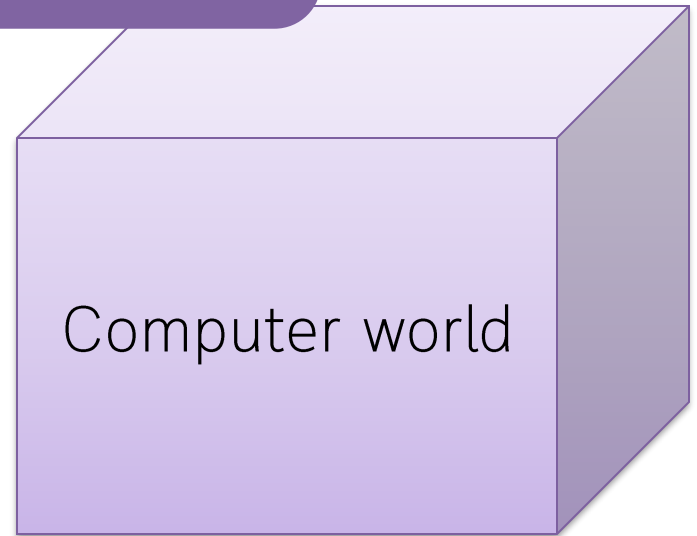
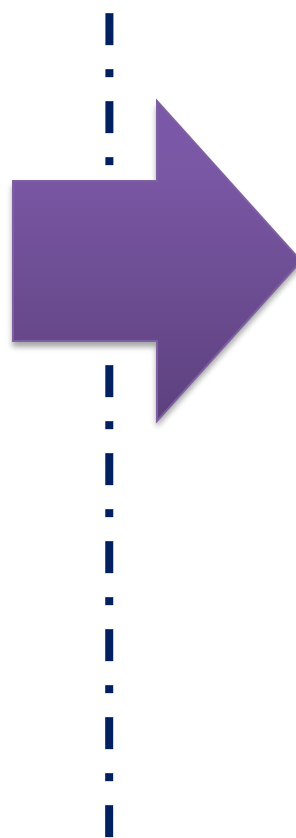


Visibility/Accessibility

จง อธิบาย information hiding ของ รถไฟฟ้า

# Abstractions

การให้ความคิดรวบยอด กับวัตถุที่มี  
อยู่ในโลกของความเป็นจริงเพื่อให้เข้าสู่  
โลกของคอมพิวเตอร์



# Tangible Object



# Intangible Object

- กฎเกณฑ์ หรือ กฎหมาย (ที่ไม่ใช่รูปเล่ม)
- ราคาสินค้า
- อากาศ
- เวลา
- ความรู้ , ทักษะต่าง ๆ
- ภาพยนตร์ (ตัวเนื้อหาของเรื่อง)
- ฯลฯ

## 2.1 Object Orientation คืออะไร (ต่อ..)

- เราได้ทราบแล้วว่า ในโลกของเรามี Object ต่าง ๆ มากมาย สิ่งที่เกิดขึ้นจาก Object ต่าง ๆ ก็คือ กิจกรรม(Activities) ความเคลื่อนไหว (Movement) หรือการกระทำ (Action ) เช่น
  - คนรับประทานอาหาร
  - สุนัขเล่นกับแมว
  - นักศึกษาเล่นเกมออนไลน์
  - นักกีฬาทีมชาติไทยลงแข่งขันกีฬาชกมวยในการแข่งขัน
  - อาจารย์สอนนักศึกษา
- หากพิจารณาโดยละเอียดแล้วจะพบว่า กิจกรรมต่าง ๆ ที่เกิดขึ้นในชีวิตประจำวันของเรานั้นล้วนแต่เกิดจากการมีความสัมพันธ์ (Relationship) และการมีปฏิสัมพันธ์ (Interaction) ระหว่าง Object 2 ตัวขึ้นไป ซึ่งจากข้อความตัวอย่างข้างต้นจะพบว่า

จะรู้ได้อย่างไรว่าอะไร  
คือวัตถุ ?

# Implement with java

```
class person {  
    String name;  
    int age;  
    String gender;  
    void eat(Fruit f) {  
        System.out.println("Eating a fruit " + f);  
    }  
}
```

ทำไมไม่มี ที่อยู่ อีเมล โทรศัพท์ ?

```
class testPerson {  
    Public static void main(String args[]) {  
        Person p1;  
        Fruit a;  
        P1.eat(a);  
    }  
}
```

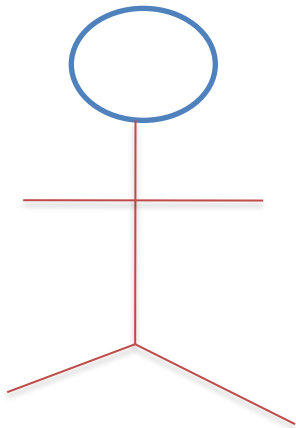
# Problem Domain

- ในโลกนี้มีวัตถุ ต่าง ๆ มากมาย และมีกิจกรรมหรือปฏิสัมพันธ์ที่เกิดขึ้นจากวัตถุ มากมาย เช่นกัน ดังนั้นเราจึงไม่สามารถสนใจทุก ๆ ทุกอย่างในเวลาเดียวกันได้ จึงจำเป็นต้องให้ความสำคัญหรือสนใจเพียงบางสิ่งบางอย่าง
- สิ่งที่เราสนใจ
- ขอบเขตของปัญหาที่เราสนใจ

กระดาษไม่พอ  
หน้าจอไม่พอ

## Attributes ของคน

- ชื่อ
- เพศ
- อายุ
- ร่างกาย
- น้ำหนัก
- ส่วนสูง
- ที่อยู่



Computer World

Real World

**0 คะแนน**

Real World

ถ้าคนนั้นคือ ลูกค้า ใน ระบบงานธนาคาร

Attribute คนในระบบธนาคาร ...?

พนักงาน ธนาคาร จะ สนใจ ร่างกาย ทำอะไร ?  
เก็บประวัติ ต้องเก็บว่ามีแขน กี่แขน มีขาที่ขา มีตาที่ตา ?

Attribute คนในระบบธนาคาร ...?

ลูกค้า หรือ พนักงาน  
ชื่อ นามสกุล ที่อยู่ เงินเดือน ฯลฯ



# Object Orientation

- Object = วัตถุ
- Orientation = การนำไปสู่
- รวมกันหมายถึง สิ่ง/**การนำไปสู่เชิงวัตถุ**

# Fruit → फल

```
• class fruit {  
    private Color color;  
    private double weight;  
    private String name;  
    private Seed seed;  
    public void eatable() {  
  
    }  
    public void setName(n) {  
        name =n;  
    }  
    public String getName(){  
        return name;  
    }  
}
```


```
class TestFruit {  
    public static void main {  
        fruit f;  
        fruit k;  
        f = new fruit();  
  
        f.setName("Apple");  
    }  
}
```

# รถยนต์

- ล้อ
- ฝา
- รุ่น
- ราคา
- ตัวถัง
- เครื่องยนต์
- ทะเบียน

VB

```
class Car
  dim c as String
  dim model as String
  dim registration as String
  dim price as double
  dim xx as .....??
end class
```



## 2.1 Object Orientation คืออะไร (ต่อ..)

- ตัวอย่าง กิจกรรมคนรับประทานอาหาร เกิดจาก Interaction “รับประทาน” ระหว่าง คนและอาหาร และเกิดจาก Relationship “เป็นเจ้าของ” ระหว่างคนและอาหาร ( เพราะคนเป็นเจ้าของอาหาร จึงจะสามารถรับประทานได้ )
- ตัวอย่าง นักมวยชกต่อยคู่ต่อสู้ เกิดจาก Interaction “ชกต่อย” ระหว่าง นักมวยและคู่ต่อสู้ และเกิดจาก Relationship “คู่ชก” ระหว่างคนและคู่ต่อสู้
- ตัวอย่าง รถกำลังวิ่งไปบนถนน เกิดจาก Interaction “วิ่งไปบน” ระหว่าง รถและถนน และเกิดจาก Relationship “การใช้งาน” รถและถนน

# รถยนต์กำลังวิ่งไปบนถนน

- Class
  - รถยนต์ , ถนน
- Object
  - ไม้ขีด .....
- ปฏิสัมพันธ์ (Interaction)
  - วิ่ง
- ความสัมพันธ์ (Relationship)
  - อยู่บน/การใช้งาน

รถยนต์ BMW สีแดงกำลังวิ่งไปบนถนนแจ้งวัฒนะ

# รถยนต์โตโยต้า camry กำลังวิ่งไปบนถนนแจ้งวัฒนะ

- Class
  - รถยนต์ , ถนน
- Object
  - รถยนต์โตโยต้า Camry , ถนนแจ้งวัฒนะ
- ปฏิสัมพันธ์ (Interaction)
  - วิ่ง
- ความสัมพันธ์ (Relationship)
  - อยู่บน/การใช้งาน

# อาจารย์ นัฐพงศ์ กำลังสอนนักศึกษาวิทยาการ คอมพิวเตอร์ รุ่น 63 วิชา OOAD

- Class
  - อาจารย์ นักศึกษา รายวิชา คอมพิวเตอร์
- Object
  - ~~—~~อาจารย์ นัฐพงศ์ , นักศึกษาคอมพิวเตอร์ รุ่น 63 , วิชา OOAD
- ปฏิสัมพันธ์ (Interaction)
  - สอน
- ความสัมพันธ์ (Relationship)
  - เจ้าของรายวิชา
  - เป็นลูกศิษย์อาจารย์

# อาจารย์บัณฑิต สอนวิชา OOAD

- Class
- -อาจารย์ , วิชา
- Object
- - บัณฑิต OOAD
- Interaction
  - สอน
- Relationship
  - อาจารย์เจ้าของวิชา




# รถยนต์ Civic ของนายวัชรวุฒีมักาลังวิ่งไป บนถนนแจ้งวัฒนะ

- Class
  - รถยนต์ คน ถนน
- Object
  - วัชรวุฒีมักาลัง , แจ้งวัฒนะ , civic
- Interaction
  - รถวิ่งบนถนน
- Relationship
  - อยู่บนถนนแจ้งวัฒนะ , เป็นเจ้าของ

# นายสมชายใช้โทรศัพท์ เพื่อเล่นไลน์

- Class
- ~~สมชาย~~ โทรศัพท์ คน แอปพลิเคชัน
- Object
  - สมชาย ไลน์
- Interaction
  - ใช้ , เล่น
- Relationship
  - เจ้าของโทรศัพท์

# รถยนต์ Civic ของนาย วัชรวุฒิจำลองวิ่งไปบนถนนแจ้งวัฒนะ

- Class
    - ถนน , **ถนน** , รถยนต์
  - Object
    - วัชรวุฒิจำลอง , แจ้งวัฒนะ , civic
  - Interaction
    - วิ่งไปบนถนน
  - Relationship
    - อยู่บนถนน , **เจ้าของ**
- 

## ต.ย. ที่ 2.1

“นาย ก. เปิดตู้เย็นยี่ห้อ A (ซึ่งเป็นของนาย ก. เอง) แล้วหยิบน้ำ (ซึ่งอยู่ในตู้เย็น) มาดื่ม”

- Object ที่เราสนใจในที่นี้ได้แก่ นาย ก. , ตู้เย็นยี่ห้อ A , และน้ำ
- Relationship ระหว่าง Object ที่เราสนใจ ได้แก่
  - นาย ก. เป็นเจ้าของตู้เย็นยี่ห้อ A
  - น้ำอยู่ในตู้เย็นยี่ห้อ A
- Interaction ระหว่าง Object ที่เราสนใจได้แก่
  - นาย ก. เปิดตู้เย็นยี่ห้อ A
  - นาย ก. หยิบน้ำ
  - นาย ก. ดื่มน้ำ

*นายปราโมทย์*เดินไปซื้อพืชชำที่เซเว่นที่อาคารบูรณาการ  
*ปราโมทย์*เดินไปซื้อยาที่เซเว่นที่อาคารบูรณาการ

- วัตถุ /คลาส
  - .....อาคาร ...อาคารบูรณาการ
  - พืชชำ
  - ยา
- Relationship
  - ..... อยู่ในอาคาร
- Interaction
  - ...เดินไป , ซื้อ

# *ปราโมทย์* เดินไปซื้อพืชชำที่เซเว่นที่อาคารบูรณาการ

## *ปราโมทย์* เดินไปซื้อถุง กับ ยางที่เซเว่นที่อาคารบูรณาการ

- วัตถุ / คลาส
  - Class ..... ลูกค้า , ถุง , ยาง , อาคาร , ร้านค้า
  - Object .... ปราโมทย์ , เซเว่นที่อาคารบูรณาการ
- Relationship
  - .....ลูกค้า ของร้านค้า
  - .... อยู่ในอาคาร
- Interaction
  - ...เดินไปซื้อ

## นางพัชรีป่วยเป็นโรค Covid-19 ไปหาหมอที่ รพ. จุฬา

- Class
  - .....หมอ , โรงพยาบาล, ผู้ป่วย , โรค
- Object
  - .....พัชรี , โรค Covid-19 , รพ. จุฬา
- Interaction
  - ไปหาหมอ , ป่วยเป็นโรค
- Relationship
  - คนไข้ ของหมอ , หมอประจำโรงพยาบาล

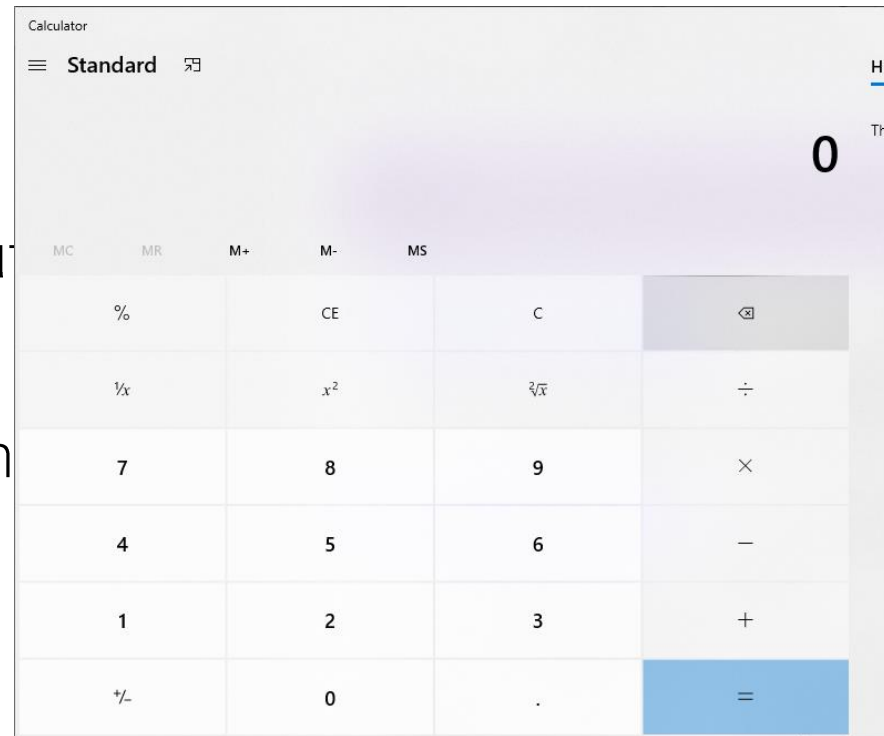
# Relationship & Interaction

- **Relationship** คือ ความเกี่ยวข้องกันหรือความสัมพันธ์ระหว่าง Object 2 ตัวขึ้นไป เช่น แม่-ลูก  
ความเป็นเจ้าของการมีอยู่ การใช้งาน เป็นต้น
- **Interaction** คือ ปฏิสัมพันธ์หรือการกระทำใด ๆ ที่เกิดขึ้นระหว่าง Object 2 ตัวขึ้นไป เช่น การสร้าง การเปลี่ยนแปลง การเล่น การกระตุ้น เป็นต้น ซึ่ง Interaction นี้เองที่ทำให้เกิด *กิจกรรม (Activities)* ต่าง ๆ ในโลกนี้



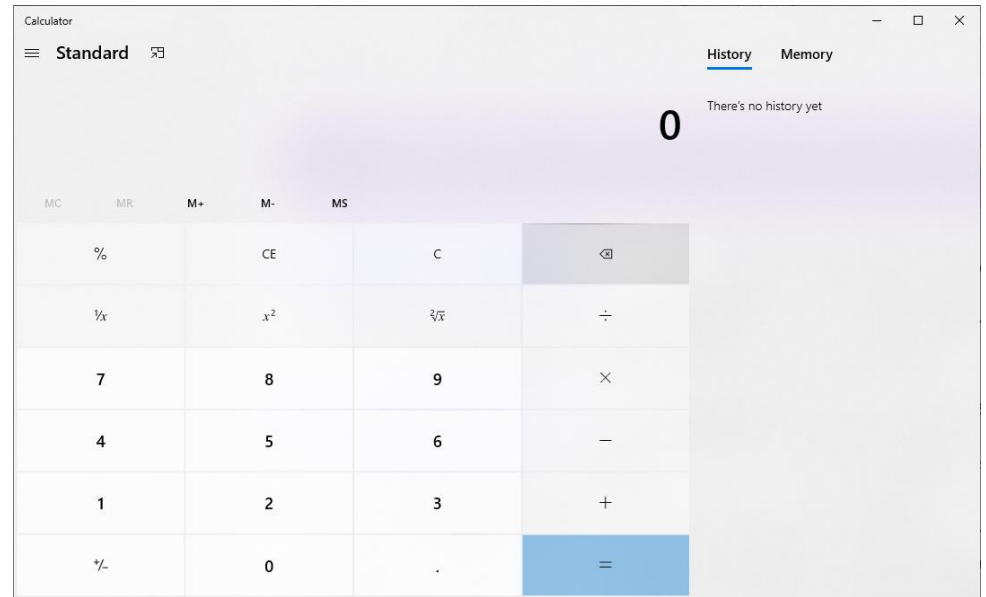
# calculator

- เป็นซอฟต์แวร์เชิงวัตถุ
  - Class
    - ฟอน์ม , Label, Textbox, Button
  - Object
    - LbResult, Button1-Button
  - Interaction
    - Load , Click , Enter
  - Relationship
    - อยู่ในฟอน์ม



# calculator

- เป็นซอฟต์แวร์เชิงวัตถุ
  - Class
    - โปรแกรมเครื่องคิดเลข
  - Object
    - เครื่องคิดเลข
  - Interaction
    - คณิตเลข คำนวณ
  - Relationship
    - โปรแกรมสำหรับคิดเลข



## 2.2 Problem Domain

- จากตัวอย่างที่ 2.1 ถ้าสังเกตให้ดีจะเห็นว่า เราจะพูดถึง Object ที่เราสนใจ หรือ Relationships ที่เราสนใจ เป็นต้น คำว่า “ที่เราสนใจ” เป็นการให้ กรอบของสิ่งที่เราต้องการพิจารณาหรือสนใจ เพราะ เราไม่สามารถสนใจในทุก ๆ วัตถุในโลกในเวลาเดียวกันได้ และในขณะเดียวกันเราก็ไม่สามารถให้ความสนใจกับทุกความสัมพันธ์ และทุก ๆ กิจกรรมหรือการกระทำที่เกิดขึ้นได้ เช่นกัน จากตัวอย่างที่ 2.1 จะเห็นว่าเราไม่สนใจสิ่งอื่น ๆ ในตู้เย็น เช่น เราไม่สนใจว่ามีผลไม้อยู่ในตู้เย็นหรือไม่ และในขณะเดียวกันเราไม่ได้สนใจว่า นาย ก. ได้ดื่มน้ำที่อยู่ในภาชนะประเภทใด เช่น อยู่ในแก้ว หรืออยู่ในขวดพลาสติก หรือ สิ่งที่อยู่ในตู้เย็น อาจจะมีสิ่งอื่น ๆ มากมาย ที่ไม่ได้พูดถึง

# มี object ที่อย่าง

- อาจารย์ สมหญิง แก่เทรด I ให้กับ นศ. สมชาย ในวิชาวิเคราะห์และออกแบบระบบเชิงวัตถุ

- ภาคเรียนใด
- ที่คน
- มหาวิทยาลัย
- ปีไหน
- สาขาวิชาอะไร

ไม่ได้อยู่ใน Problem Domain

โจทย์ ไม่ได้กำหนดไว้

**“ไม่มี ธนาคารไหน หรือ บ. มหาชนใด ๆ  
จ้าง นศ. ไปเขียน HelloWorld”**

## แบบฝึกหัด

ข้อ 1. ให้บอกว่่าสิ่งใดต่อไปนี้เป็น Class, object , Interaction ,Relationship

1. นายสมชาย กำลัง เขียนโปรแกรมด้วยภาษา java
2. สมหญิง กำลังถ่ายเอกสาร
3. สมศรีกำลัง Line หาแฟนชื่อ ณัฐวุฒิ ด้วยโทรศัพท์มือถือยี่ห้อ Samsung รุ่น s20
4. นาย สมควร ไปถอนเงิน จำนวน 5000 บาทที่ตู้ ATM ของธนาคารออมสิน

ทำลงกระดาษ ส่งทางไลน์ สร้างเป็นโน้ต ใส่ชื่อ รหัส 13 หลัก  
ส่งก่อน 20.00 น.

## 2.2 Problem Domain (ต่อ)

- ใน Problem Domain หนึ่ง นั้นสามารถมี Object ได้ตั้งแต่ 2 ตัวขึ้นไป จนกระทั่งนับไม่ถ้วน

## 2.2.1 Object และ Classes

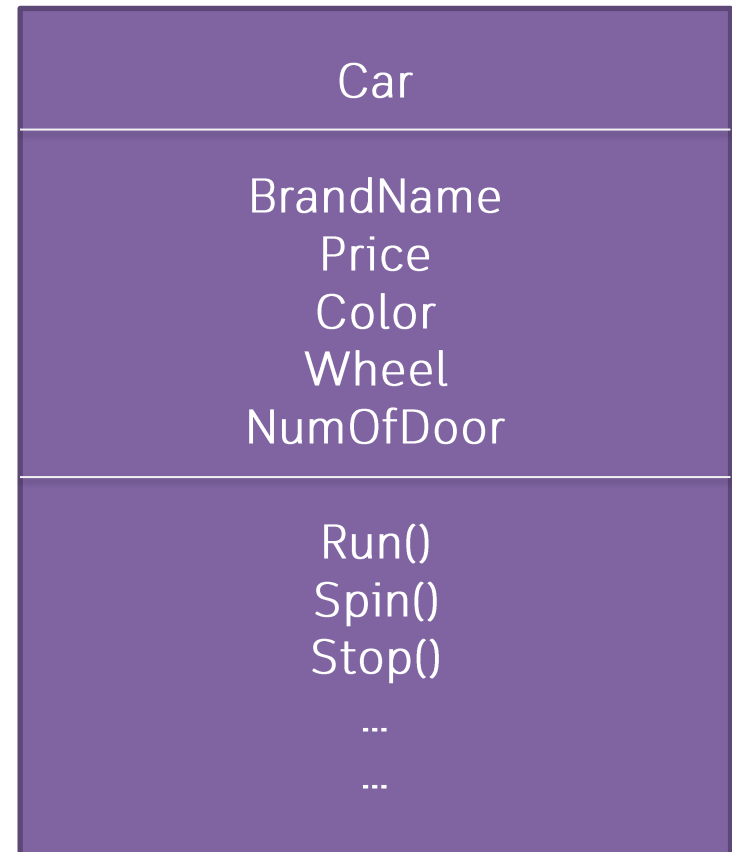
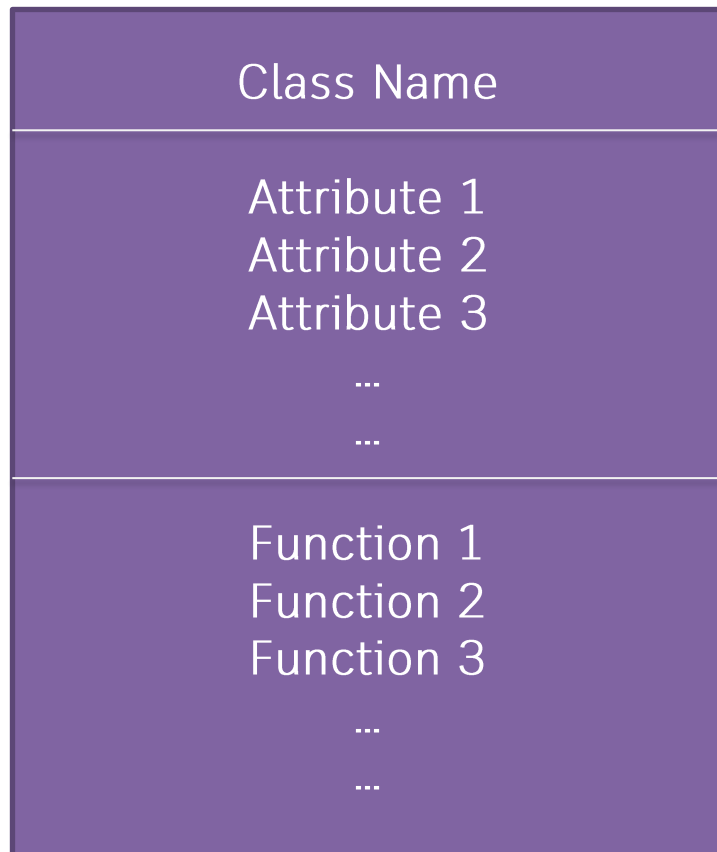
- Concept ?
  - ที่ผ่านมามองแต่ Object ที่อยู่ใน Domain ที่มีอยู่จริงในโลก (Real World) เท่านั้น และถ้าเราต้องการที่จะจำลอง (Model) สิ่งต่าง ๆ ให้อยู่ในเครื่องคอมพิวเตอร์จะทำได้อย่างไร
- Concept คืออะไร
  - Concept หมายถึง **ความคิดรวบยอดที่เราให้กับวัตถุนั้น ๆ** ( แนวความคิดในแง่ของความเท็จจริง ไม่เกี่ยวกับความรู้สึกที่เรามีต่อสิ่งนั้น )
  - ตัวอย่างเช่น ถ้าเรา ต้องการให้ Concept กับรถยนต์ นั่นคือ รถทุกคันต้องมีตัวถัง มีล้อ และมีเครื่องยนต์ หรือเมื่อต้องการให้ concept เกี่ยวกับ สุนัข ( สุนัขปกติทั่วไป ) นั่นคือ สุนัขทุกตัวต้องมี 4 ขา มีขน มีหาง สามารถเห่า และ กัดได้เหมือนกันทุกตัว

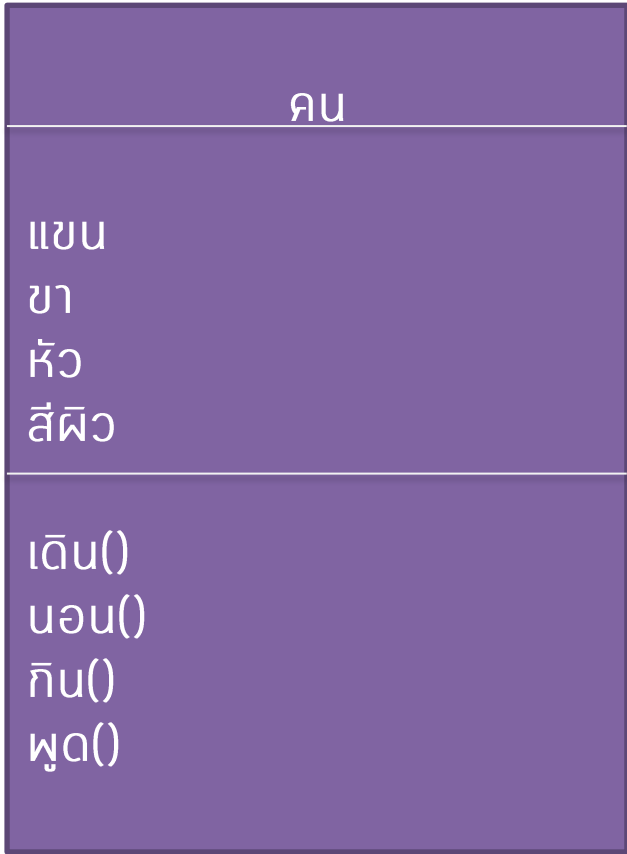
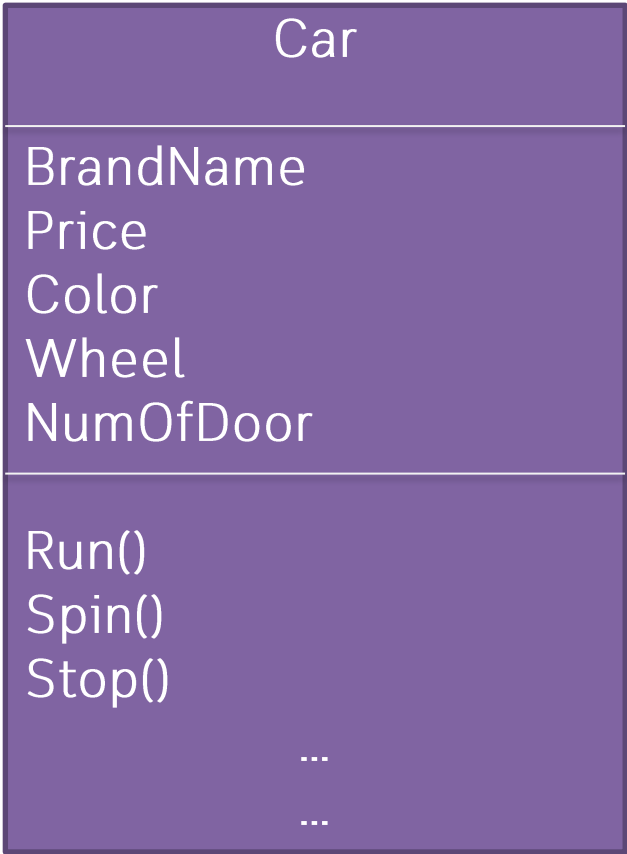


## 2.2.2 Attribute & Function

- Attribute
  - เราสามารถบรรยายคุณสมบัติของ object ต่าง ๆ ในโลกได้ โดยที่คุณสมบัติดังกล่าวนี้เป็นคุณสมบัติที่เราสนใจหรืออยู่ใน Domain ที่เราสนใจนั่นเอง เช่น สีและจำนวนประตูของรถคันหนึ่ง หรือ อาจจะเป็น สีผิวและเพศของคนคนหนึ่ง เป็นต้น เราเรียกคุณสมบัติเหล่านี้ว่า Attribute
- Function
  - ความสามารถหรือพฤติกรรมของวัตถุนั้น ๆ ที่สามารถแสดงออกมาได้

# รูปแบบภาพแสดงการอธิบาย Class





## 2.3 Object-Oriented Software Engineering (OOSE)

- OOSE หมายถึง กระบวนการทางคอมพิวเตอร์ตั้งแต่ ขั้นตอนการวิเคราะห์ การออกแบบ การพัฒนาโปรแกรม และการนำไปใช้ โดยใช้หลักการของ Object-Oriented
- *OOA :: Object - Oriented Analysis*
- *OOD :: Object - Oriented Design*
- OOP :: Object - Oriented Programming
- OO Testing

# คำถามท้ายบท

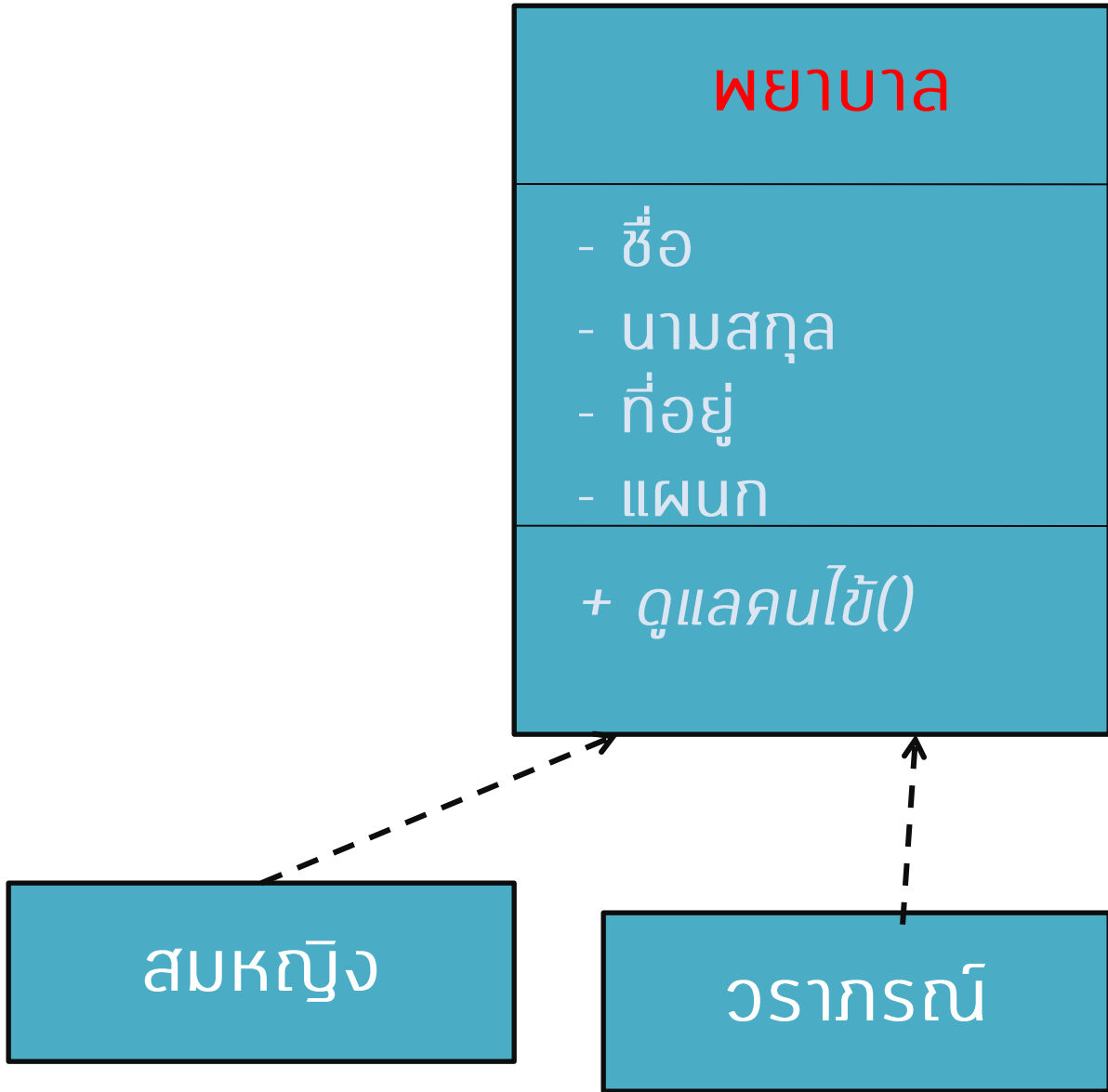
- จงแยกแยะ ว่าสิ่งใดต่อไปนี้เป็น Class และอะไรคือ Object
  - นายสมชาย ขับรถยนต์โตโยต้า Vios ไปบนถนน แจ็งวัฒนะ
    - Class : รถยนต์ ถนน คน **สมชายขับรถยนต์ //ปิด**
    - Object : สมชาย Vios แจ็งวัฒนะ
  - พยาบาล ซ้ำยั้งลักษณะฉีดวัคซีนป้องกันโรคพิษสุนัขบ้าให้กับคนไข้ชื่อ ทักซิณ
    - Class :: พยาบาล โรค คนไข้ วัคซีน
    - Object :: ซ้ำยั้งลักษณะ ทักซิณ โรคพิษสุนัขบ้า
  - ด.ช. อริสมัน สอบตกวิชาคณิตศาสตร์
  - Class : นักเรียน วิชา
  - Object : อริสมันต์ วิชาคณิตศาสตร์

# ด.ช. อริสมันสอบตกวิชาคณิตศาสตร์

- Object
  - .... ด.ช. อริสมัน และ วิชาคณิตศาสตร์
- Class
  - .... วิชา และ นักเรียน
- Interaction
  - .... สอบตก
- Relationship
  - .....

นาย สมควร ไปถอนเงิน จำนวน 5000 บาทที่ธนาคาร ออม  
สิน

- Class
- Object
- Interaction
- Relationship





## ถนน

- ชื่อ
- เลข
- ความยาว

- + บอกชื่อ
- + บอกจำนวนเลข
- + ความยาว

# คำถามท้ายบท

- จงให้ความหมายของคำศัพท์ต่อไปนี้
  - 2.1 Computer World , Real Word
  - 2.2 Problem Domain
  - 2.3 Concept
  - 2.4 Interaction
  - 2.5 Relationship

# 2.4 Abstractions

## Agenda

- Objectives : วัตถุประสงค์
- Abstractions
  - Classification Abstraction
  - Aggregation Abstraction
  - Generalization Abstraction
  - Association Abstraction
- Assignment

# Objectives

---

- เพื่อให้ นศ. เข้าใจความหมายและความสำคัญของ Abstraction ที่มีต่อ OOAD
- เพื่อให้ นศ. ทำความเข้าใจเบื้องต้นเกี่ยวกับ Abstraction แบบต่าง ๆ ได้แก่ Classification, Aggregation, Generalization และ Association
- เตรียมความรู้พื้นฐานเพื่อเข้าใจการเรียนรู้ในขั้นต่อไป

# ความหมายของ Abstractions

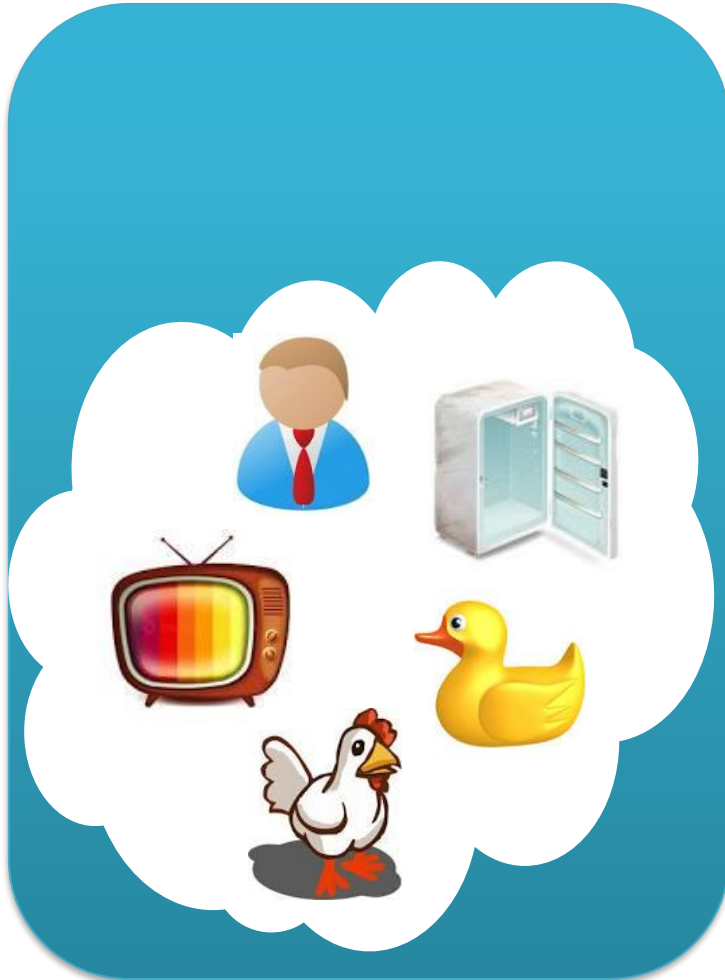
- เราได้รู้จักกับ Object และ Class ทั้งในโลกของความจริง (Real world) และในคอมพิวเตอร์
- กระบวนการในการให้ Concept แบบคร่าว ๆ กับ Object ต่าง ๆ ใน Real World เพื่อสร้าง Class นั้นเราเรียกว่า Abstractions ซึ่งแบ่งออกได้เป็น 4 กระบวนการย่อย คือ
  - Classification Abstraction
  - Aggregation Abstraction
  - Generalization Abstraction
  - Association Abstraction

# สรุปคำศัพท์

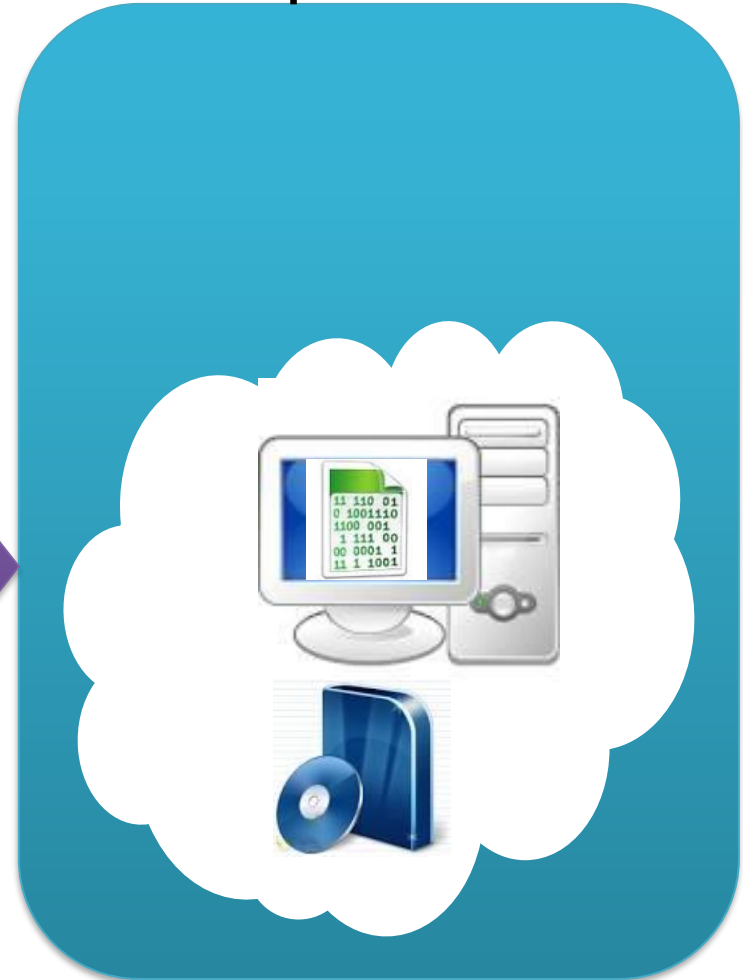
- Class
- Object
- Polymorphism / information hiding / encapsulation / Reusable /...
- Real world / Computer World
- Problem Domain
- Interaction / Relationship
- Concept /

# Concept

Real world



Computer world



# *Abstractions*

## Classification Abstraction คืออะไร

- Classification Abstraction เป็นกระบวนการที่ใช้แยกประเภท (Classify) Object ต่าง ๆ ที่อยู่ใน Domain และให้ความคิดรวบยอด (Concept) กับ Object ต่าง ๆ เหล่านั้น เพื่อให้ได้ “Class พื้นฐาน” (**Fundamental Classes**) ที่ต้องการ
- เราจะสร้างคลาสจากวัตถุต่าง ๆ ที่อยู่ใน Problem Domain



# คำถาม ถ้าหาก นศ. เดินเข้ามาในห้องเรียน

- จะเจอสิ่งใดบ้าง
- แบ่งออกเป็นกี่กลุ่ม อะไรบ้าง
  - อะไรคือ class
  - อะไรคือ object

คำถามนี้ คือต้องการจะรู้ว่า เวลา นศ. เดินเข้ามา แล้ว รู้จักสิ่งของ/วัตถุอะไรบ้าง?

# Summary / สรุป คำศัพท์ ที่กล่าวถึงไปแล้ว

- **Class**
- **Object**
  - **Tangible**
  - **Intangible**
- **Concept**
- **Abstraction**
- **Information Hiding**
- **Encapsulation**
- **Problem Domain**

# เวลาเข้ามา

- โต๊ะ
- เก้าอี้
- คอมพิวเตอร์
- หลอดไฟ
- ลำโพง
- ไมโครโฟน
- ผับ
- กระจก

*classes*

*obejcts*

*โต๊ะ 1 เก้าอี้ สีแดง  
โต๊ะอาจารย์  
หลอดไฟ 1  
ปลั๊กไฟ 1*

- กระดาน
- ฉาก
- โปรเจ็กเตอร์
- นาฬิกา
- แอร์
- พัดลม
- ปลั๊กไฟ

# แผนภาพแสดงความคิดรวบยอด (concept)

รถจักรยานยนต์ซูซูกิ

รถยนต์โตโยต้า

รถโดยสาร 6 ล้อ volvo

Concept เกี่ยวกับรถ

1. มีเครื่องยนต์
2. มีล้อจำนวนหนึ่ง
3. วิ่งไปบนถนนได้
4. ใช้แก๊สหรือน้ำมันเป็นเชื้อเพลิง

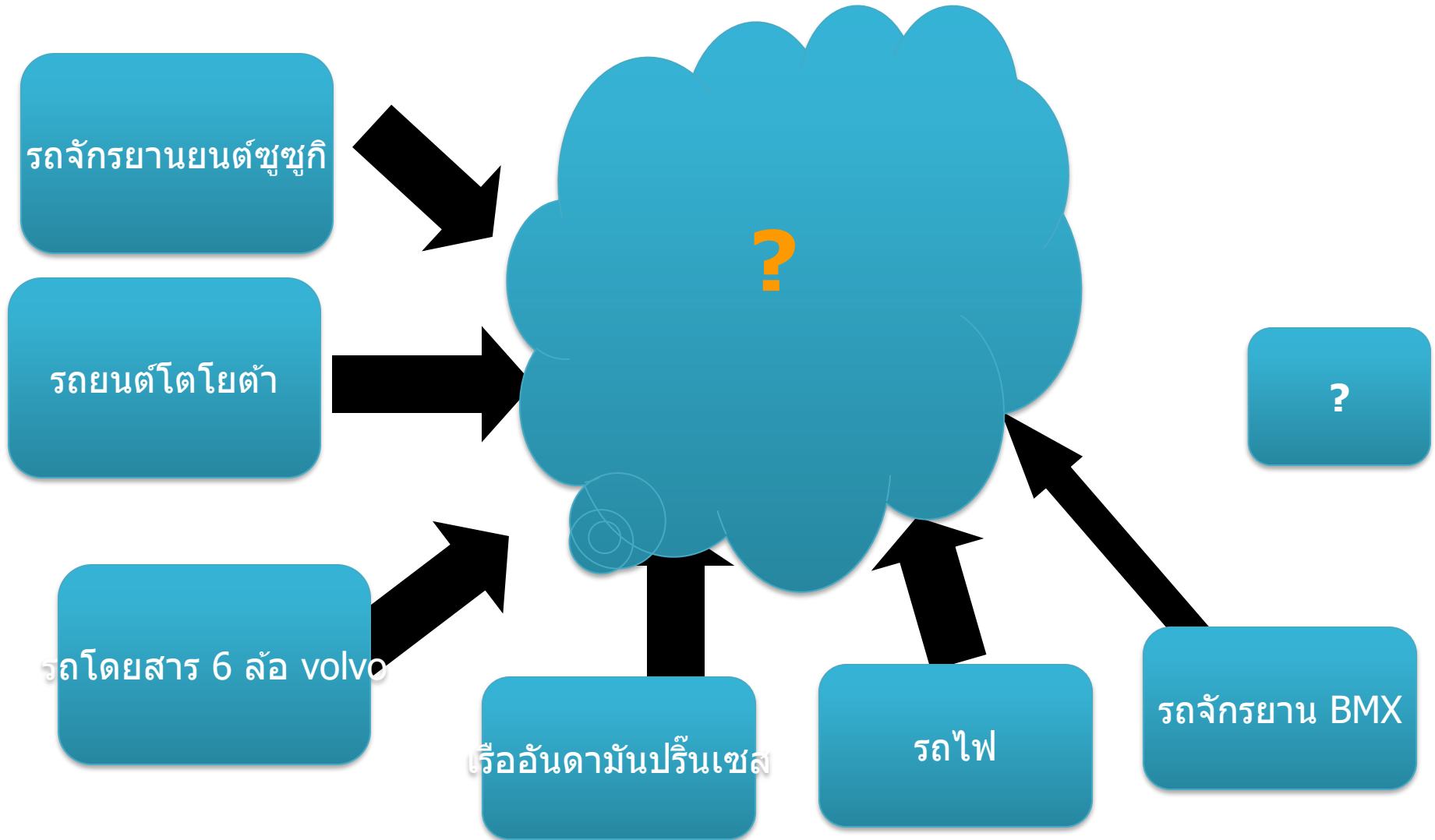
รถยนต์

เรืออันดามันปรินเซส

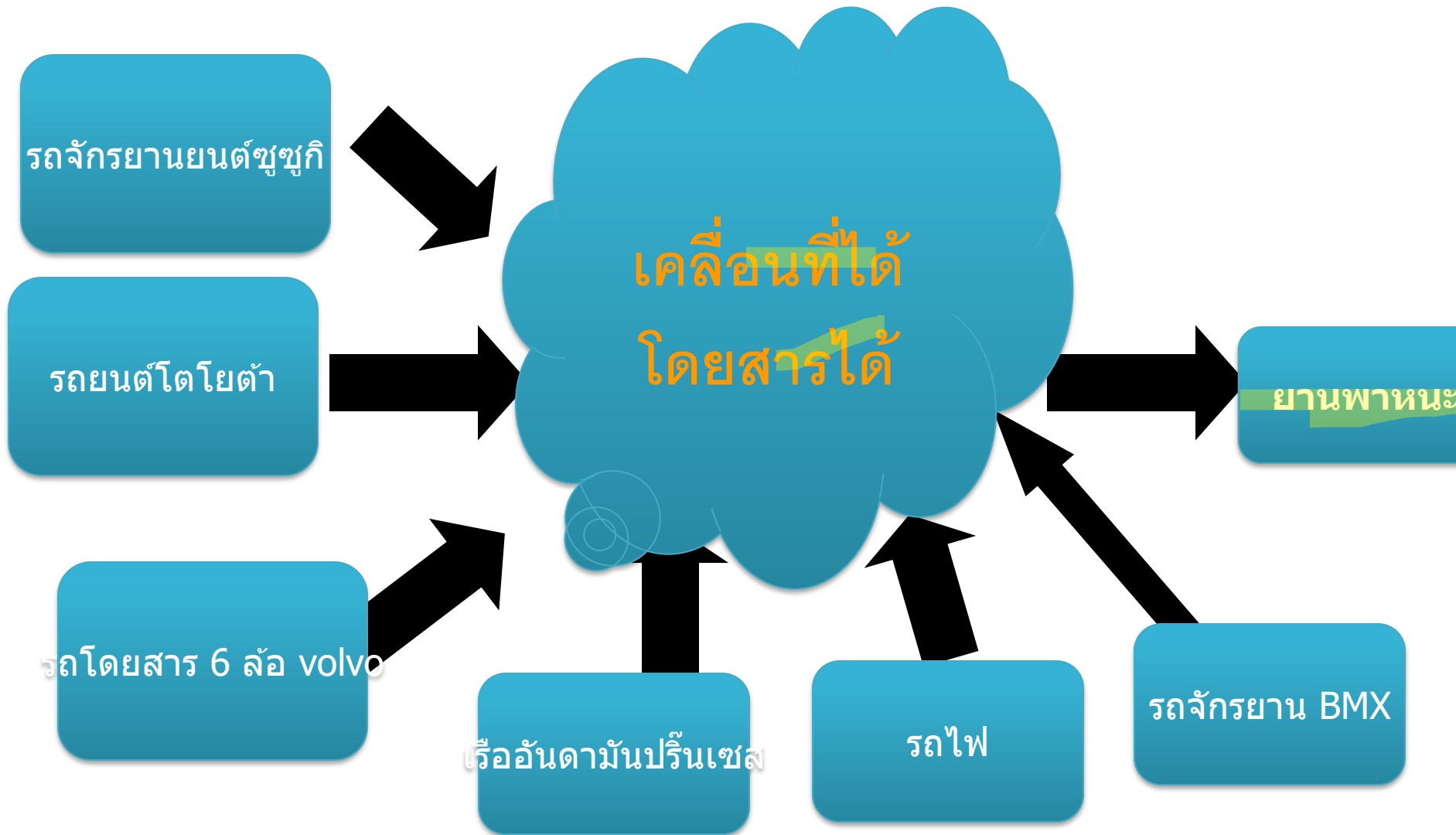
รถไฟ

รถจักรยาน BMX

# ถ้าต้องการให้ concept ใช้ได้กับทุก ๆ object ข้างล่างนี้จะต้องทำอย่างไร



ถ้าต้องการให้ concept ใช้ได้กับทุก ๆ object ข้างล่างนี้จะต้องทำอย่างไร



## แบบฝึกหัดที่ 2.2 จงวาดแผนภาพแสดงความคิดรวบยอดหรือ concept ด้านล่างนี้

2.2.1 จงให้ Concept ของ คอมพิวเตอร์ (Computer)

2.2.2 จงให้ Concept ของ นักศึกษา

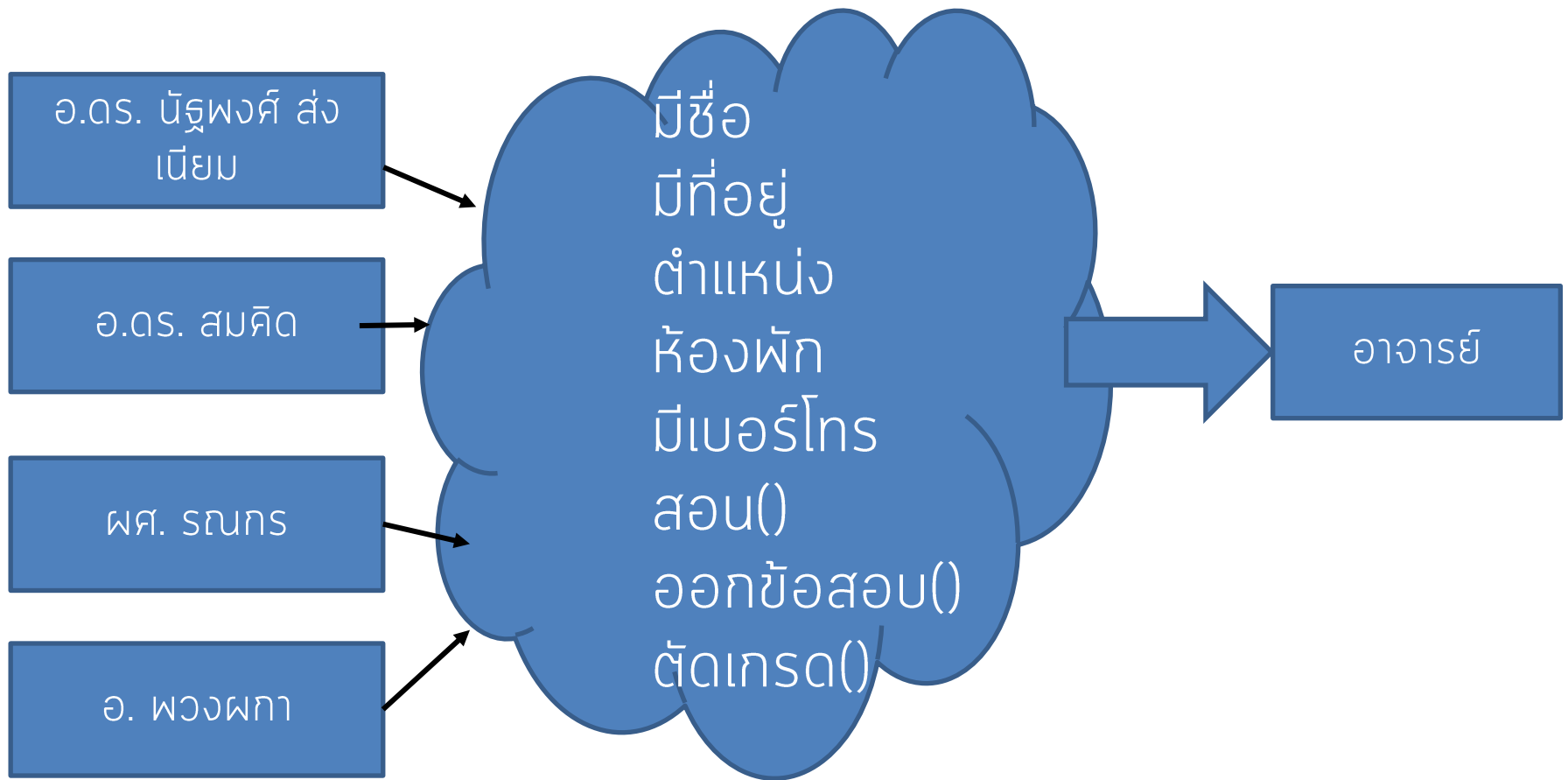
2.2.3 จงให้ Concept ของ โทรทัศน์ (TV)

2.2.4 จงให้ Concept ของ กีฬาฟุตบอล

2.2.5 จงให้ Concept ของ โทรศัพท์มือถือ

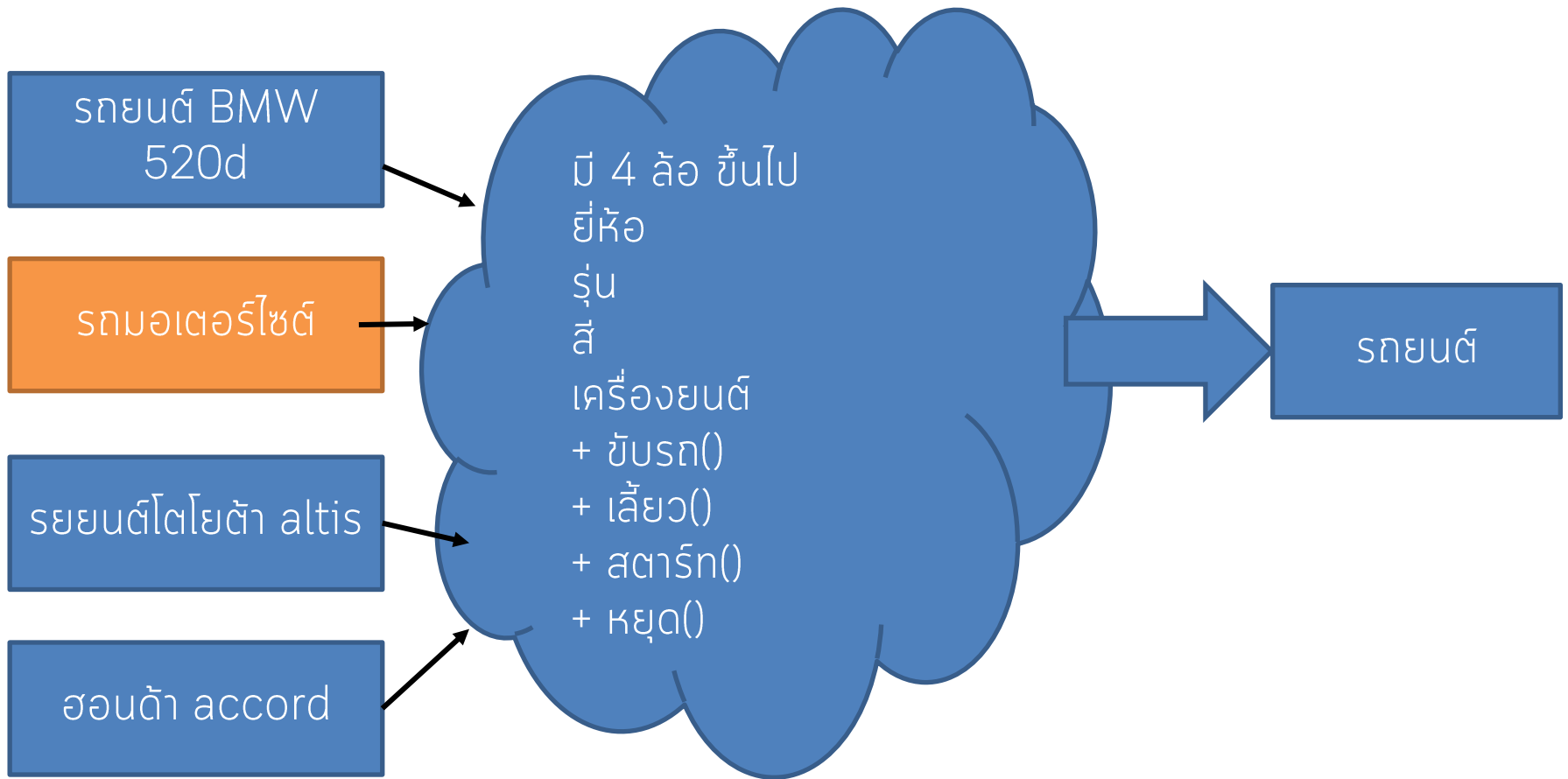
2.2.5 จงให้ Concept ของ รถยนต์

# ตย. ให้ความคิดรวบยอดของ อาจารย์





# ตย. ให้ความคิดรวบยอดของรถยนต์



# Computer

เครื่องคิดเลขไฟฟ้า



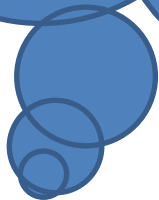
- ใช้ในการเก็บข้อมูล
- ใช้ไฟฟ้า
- ประมวลผลได้

- จอภาพ
- คีย์บอร์ด
- เมาส์



Computer

โทรศัพท์มือถือ



# Computer

HP pavilian

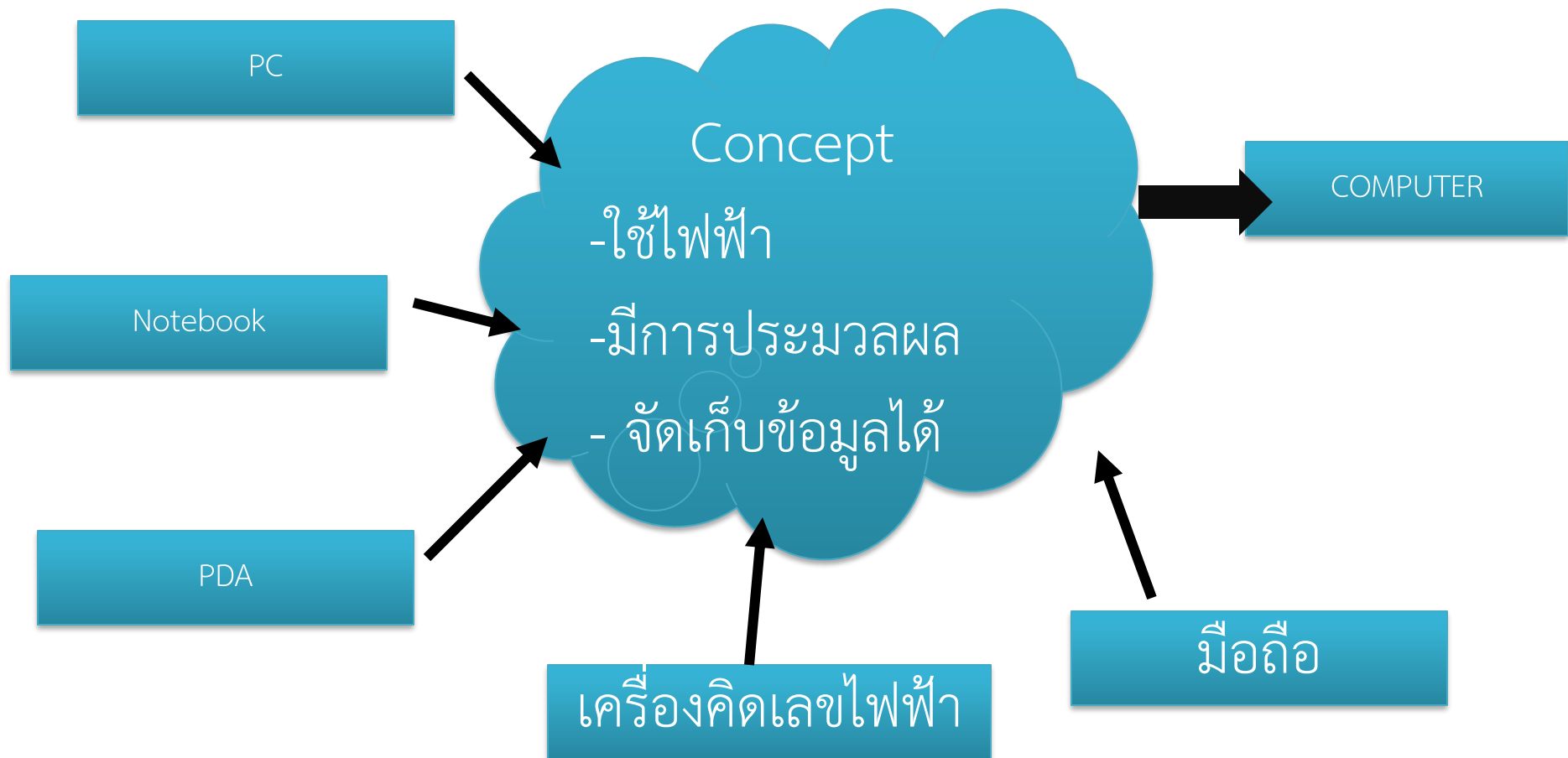
- ใช้ในการเก็บข้อมูล
- ใช้ไฟฟ้า
- ประมวลผลได้

- จอภาพ
- คีย์บอร์ด
- เมาส์

ACER spire

Computer

นศ. จะต้อง concept ที่ละเอียดหรือดีพอ ที่จะ  
ไม่ให้วัตถุที่ไม่เกี่ยวข้องเข้ามาอยู่ในกลุ่มนี้ได้



ภาคปกติ

### Concept

- ชื่อ
- รหัสนักศึกษา
- วิชาเอก
- รุ่น/หมู่
- ลงทะเบียนได้

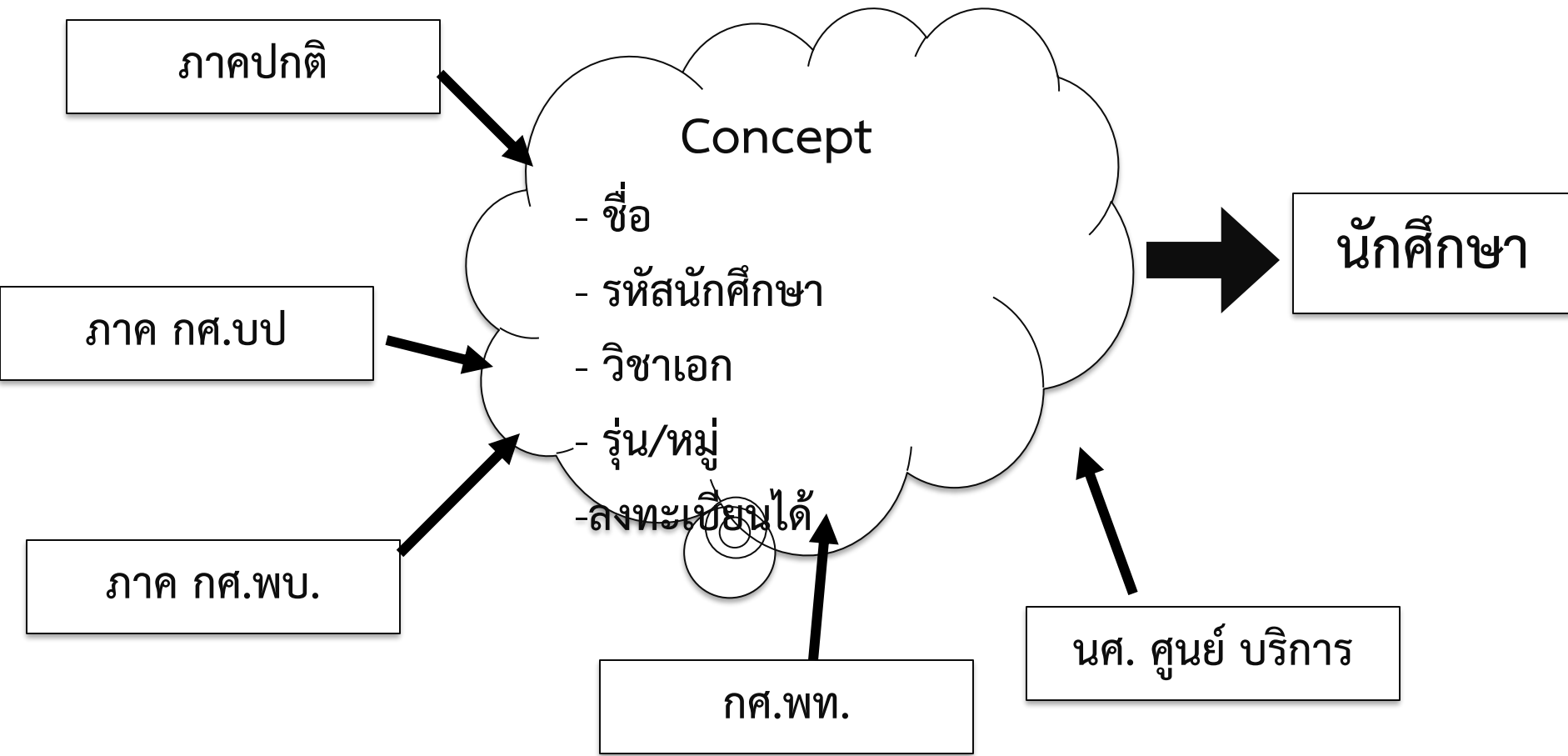
นักศึกษา

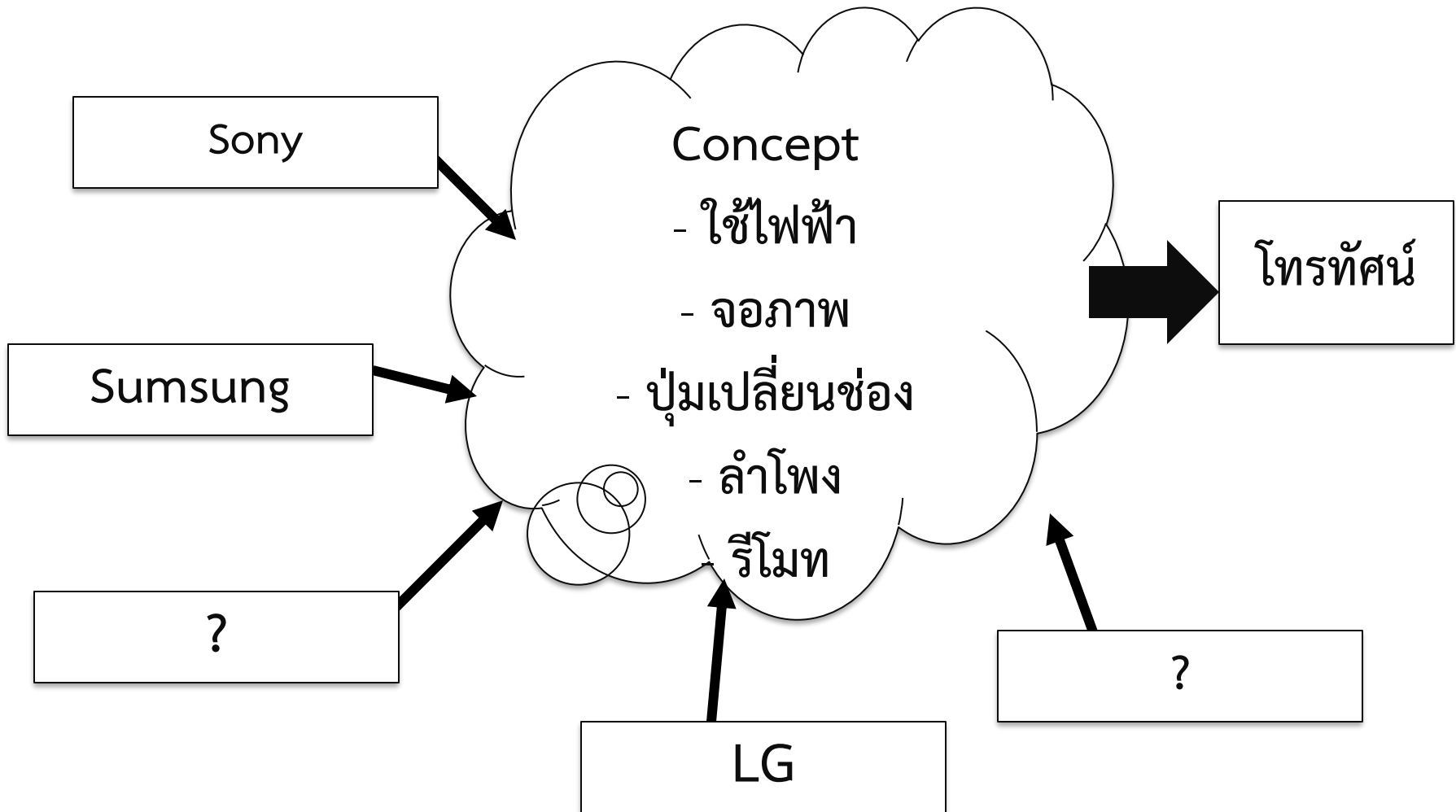
ภาค กศ.บป

ภาค กศ.พบ.

กศ.พท.

นศ. ศูนย์ บริการ





## Classification Abstraction (cont..)

เราถือว่า Classification เป็น abstraction ที่สำคัญที่สุด เพราะ Class เกิดขึ้นด้วย Classification Abstraction ถ้าหาก Class พื้นฐานที่เกิดขึ้นในขั้นตอนนี้เกิดข้อผิดพลาด การสร้าง Class ใหม่ ๆ ด้วยกระบวนการ Aggregation และ Generalization และการสร้างความสัมพันธ์ ระหว่าง Class ต่าง ๆ ด้วย Association ย่อมเกิดความผิดพลาดด้วยเช่นกัน

# Aggregation Abstraction

- Aggregation Abstraction คือ กระบวนการที่นำเอา Class พื้นฐานที่สร้างขึ้นจาก Classification Abstraction มารวมกันหรือประกอบกัน (Aggregation) เพื่อให้เกิด Class ที่ใหญ่ขึ้น หรือ ซับซ้อนขึ้น
- การทำ Aggregation คือการตอบคำถามว่า ใน Class กลุ่มหนึ่งนั้น เราสามารถที่จะนำมันมารวมกันเพื่อทำให้เกิด Class ใหม่ ที่มี Concept ใหม่ ได้หรือไม่อย่างไร หรือ อีกนัยหนึ่ง คือการตอบคำถามว่ามี Class ใดบ้างที่สามารถแบ่งออกเป็น Class ย่อยโดยมี Concept ต่างไปจากเดิมได้

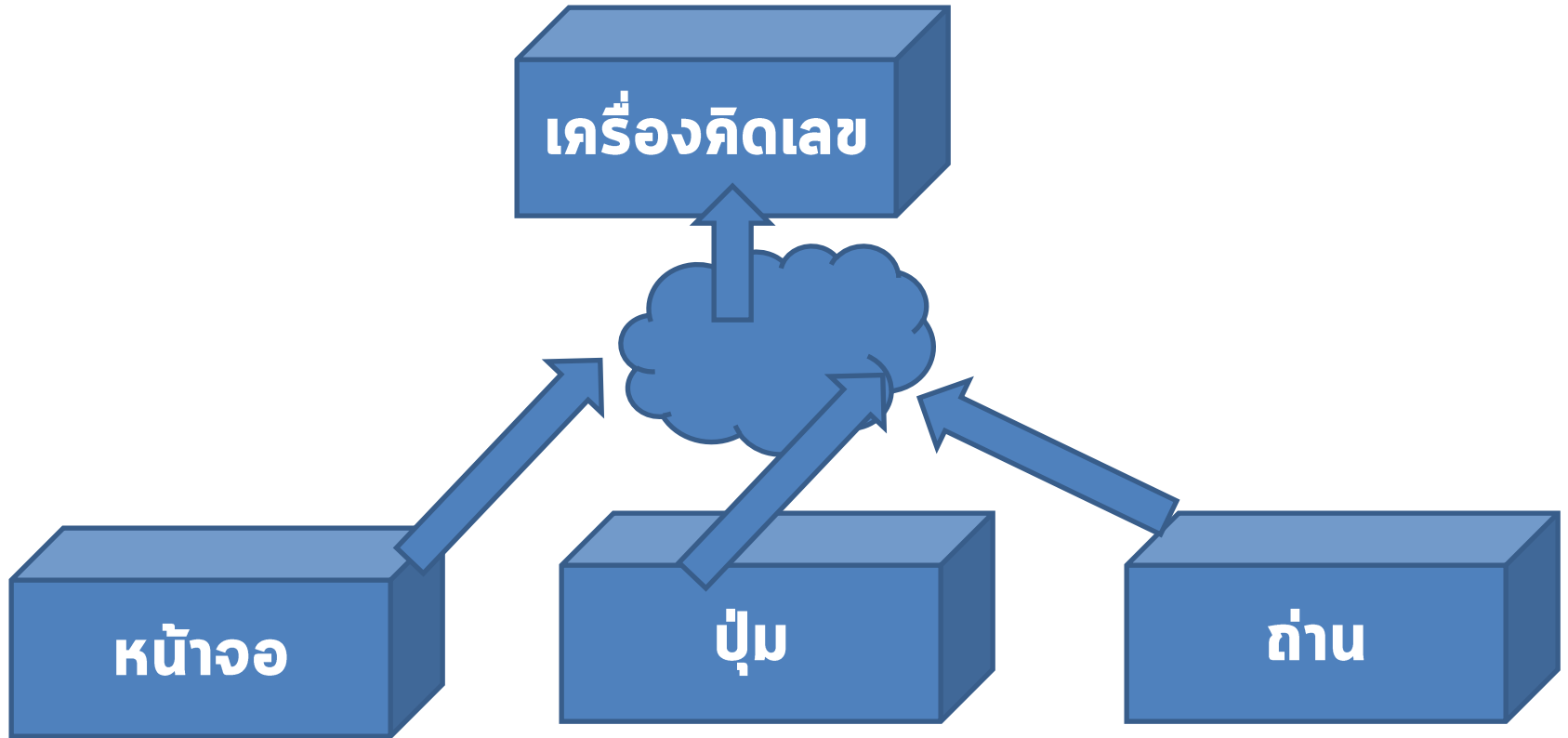


# ระบบจองห้องพักโรงแรม

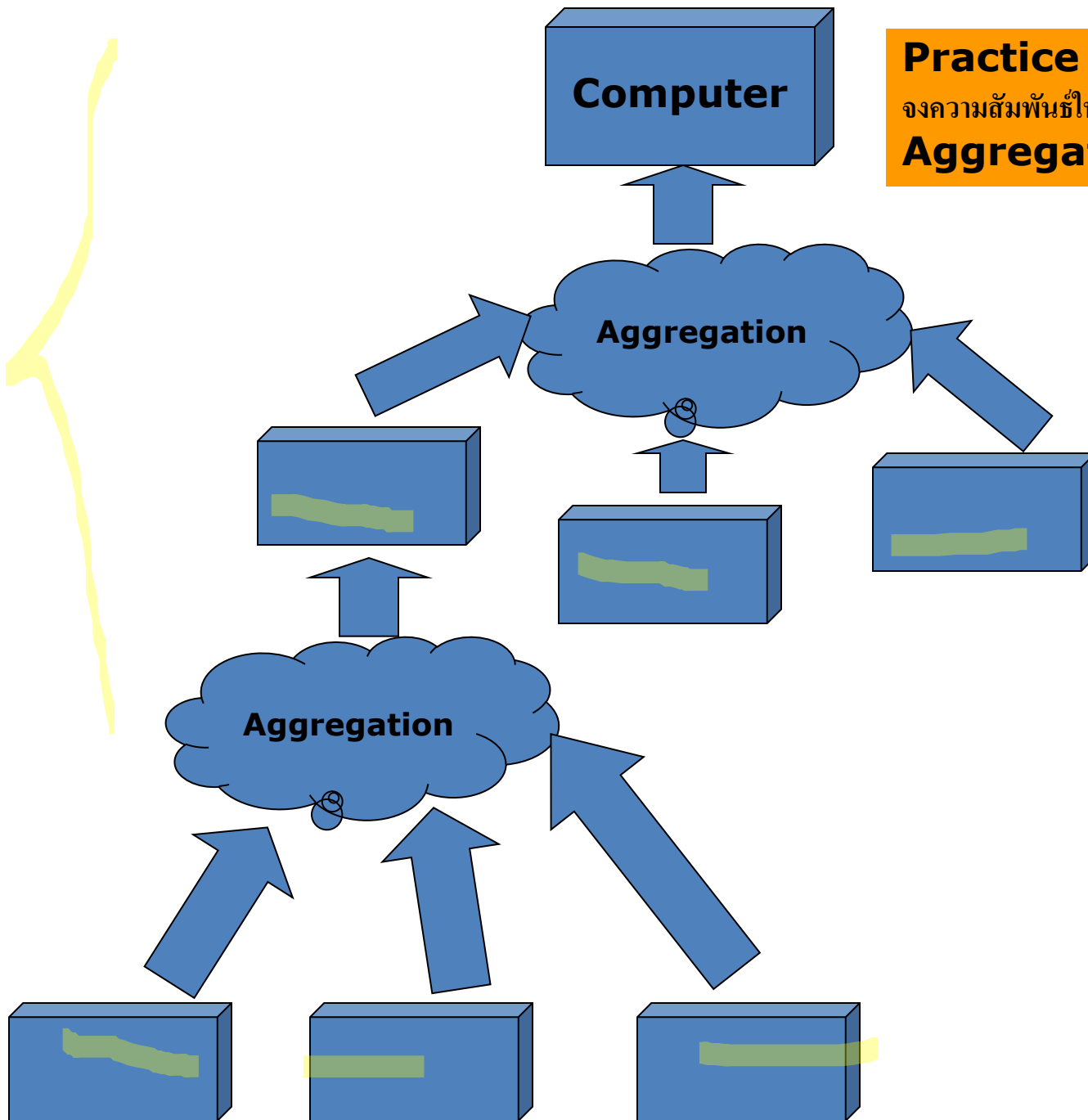
- คลาสลูกค้า
- คลาสพนักงาน
- คลาสห้องพัก
- คลาส อาหาร



# เครื่องคิดเลข



**Practice I**  
จงความสัมพันธ์ในเชิง  
**Aggregation ของ Computer**

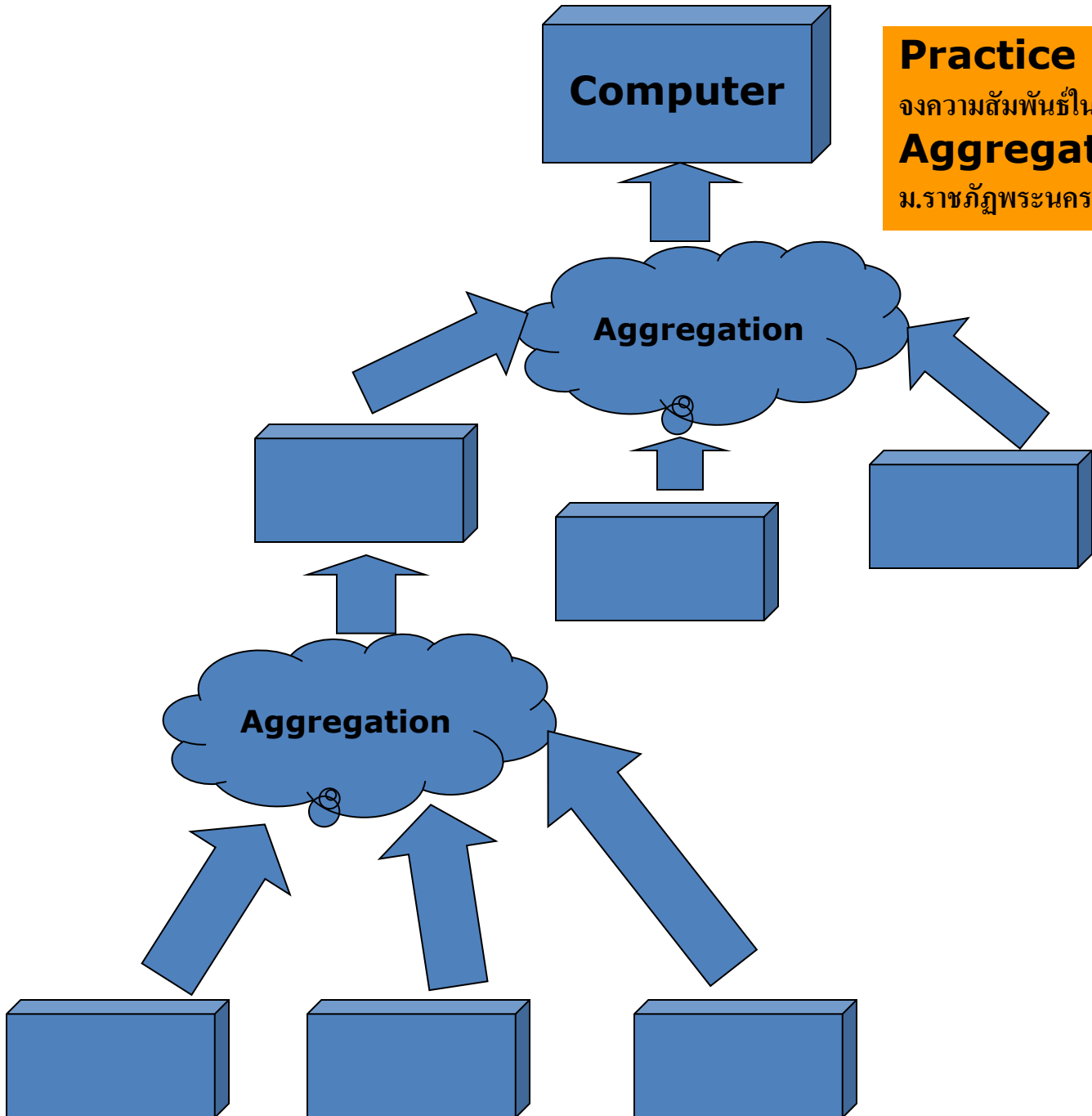


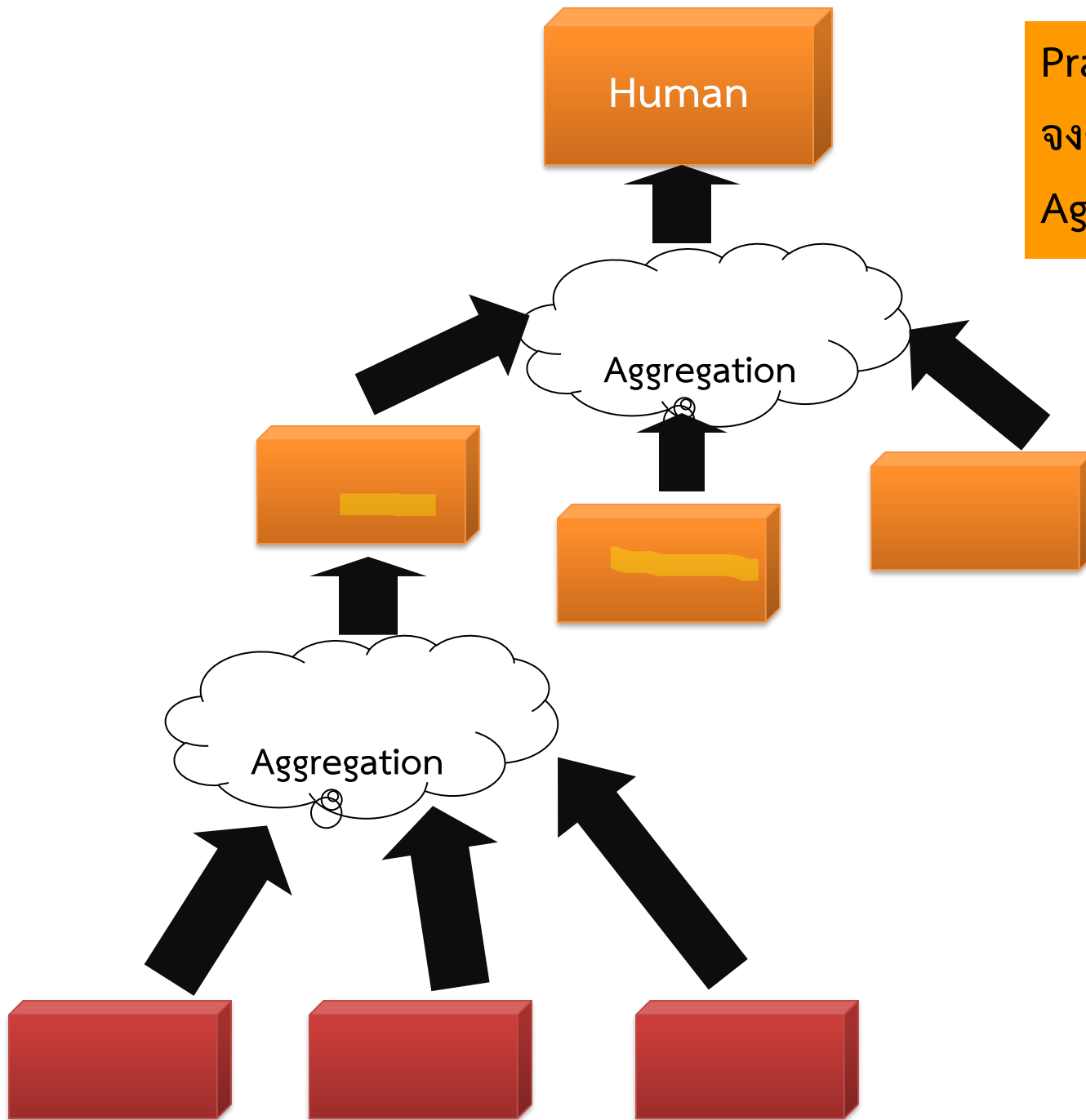
Computer

**Practice I**  
จงความสัมพันธ์ในเชิง  
**Aggregation** ของ บุคลากรของ  
ม.ราชภัฏพระนคร

Aggregation

Aggregation

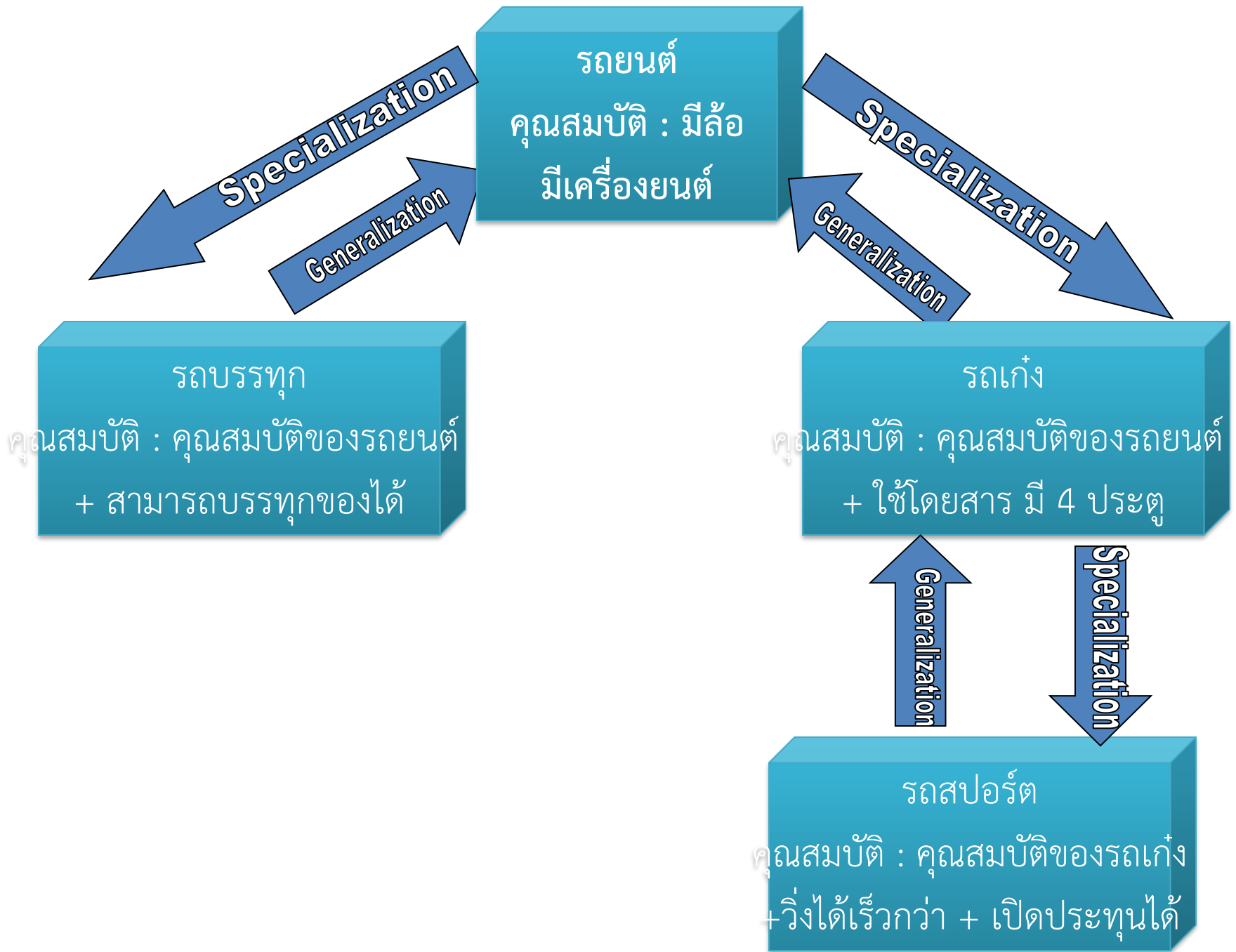




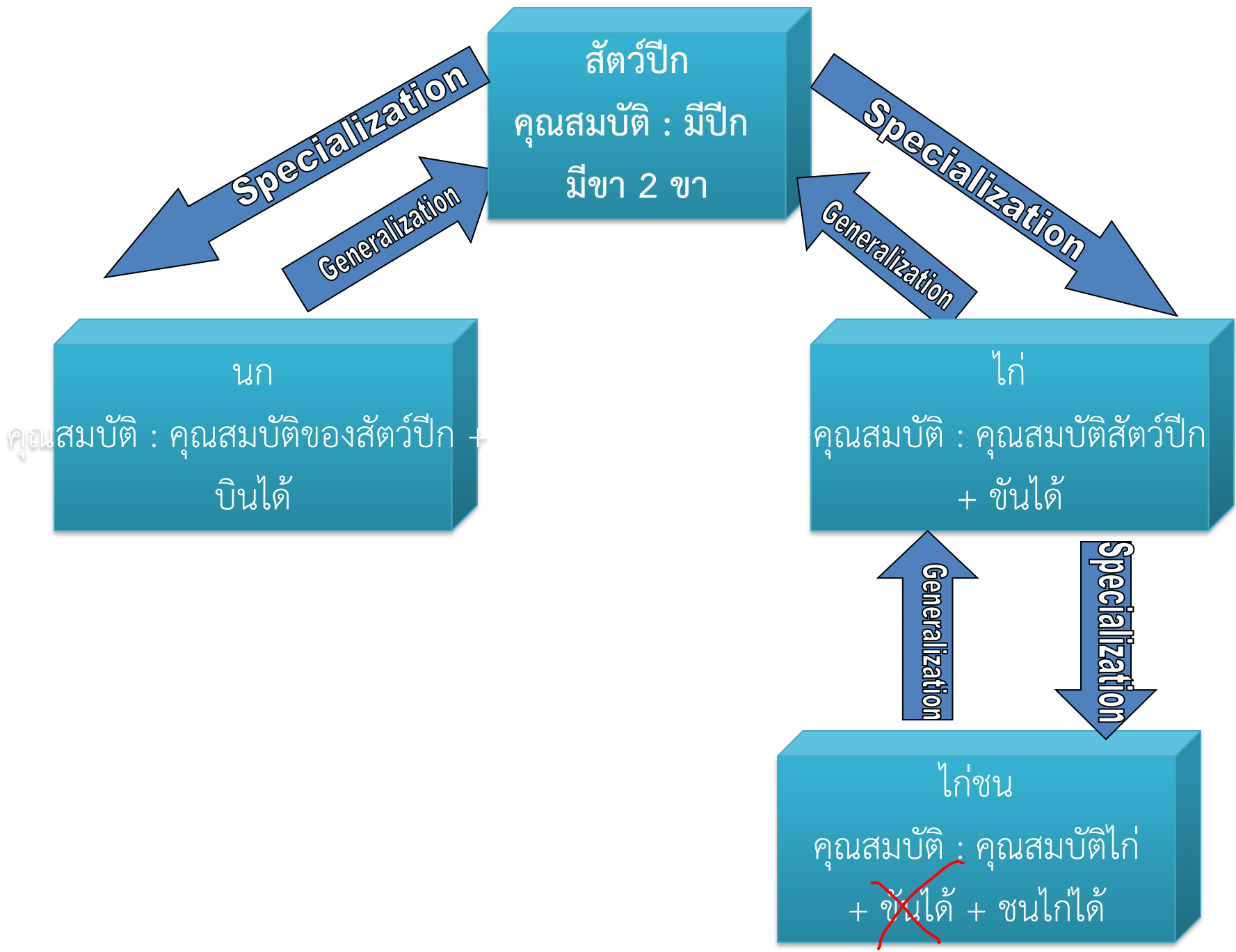
Practice I  
จงความสัมพันธ์ในเชิง  
Aggregation ของ Human

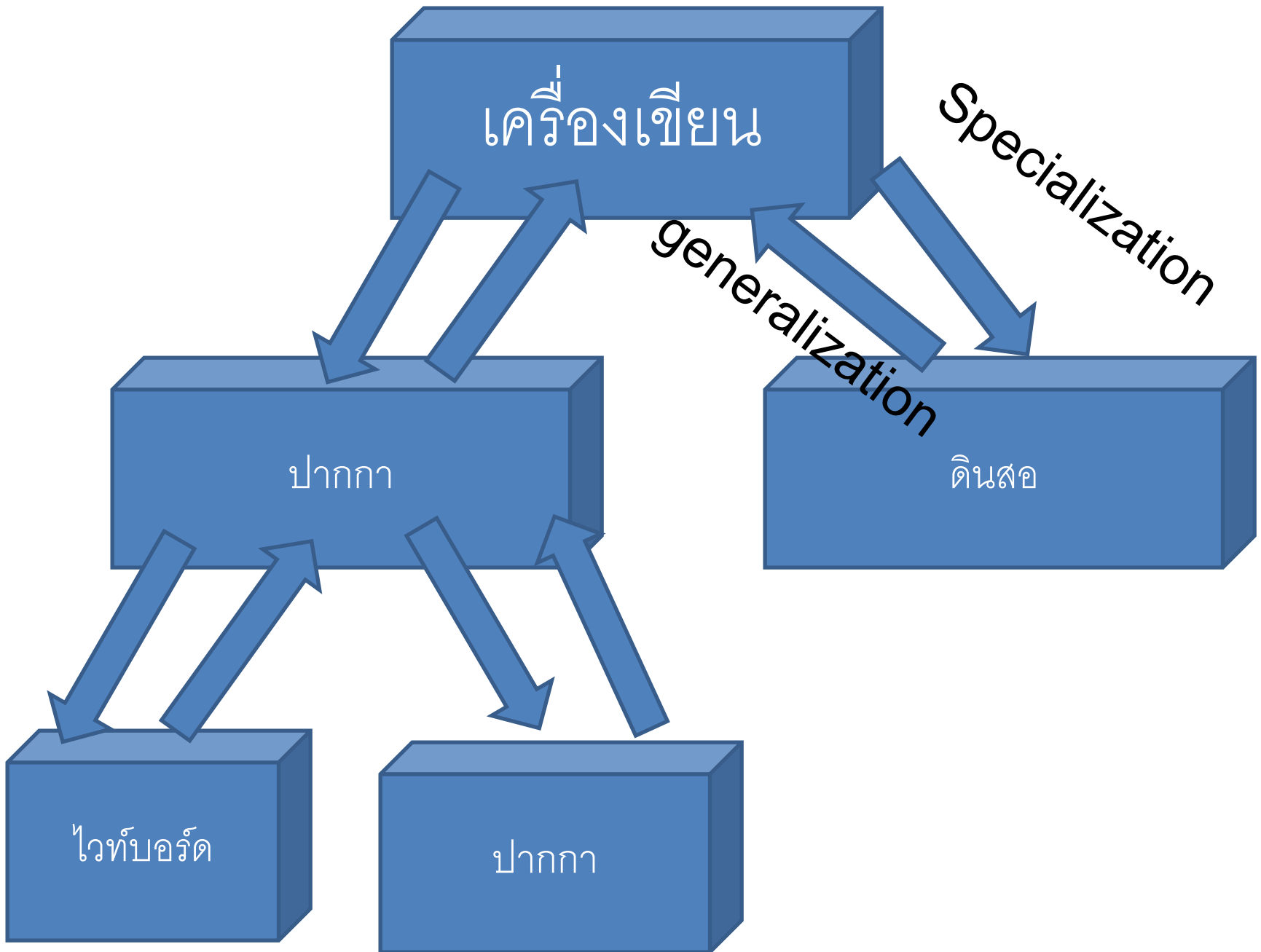
# 3. Generalization Abstraction

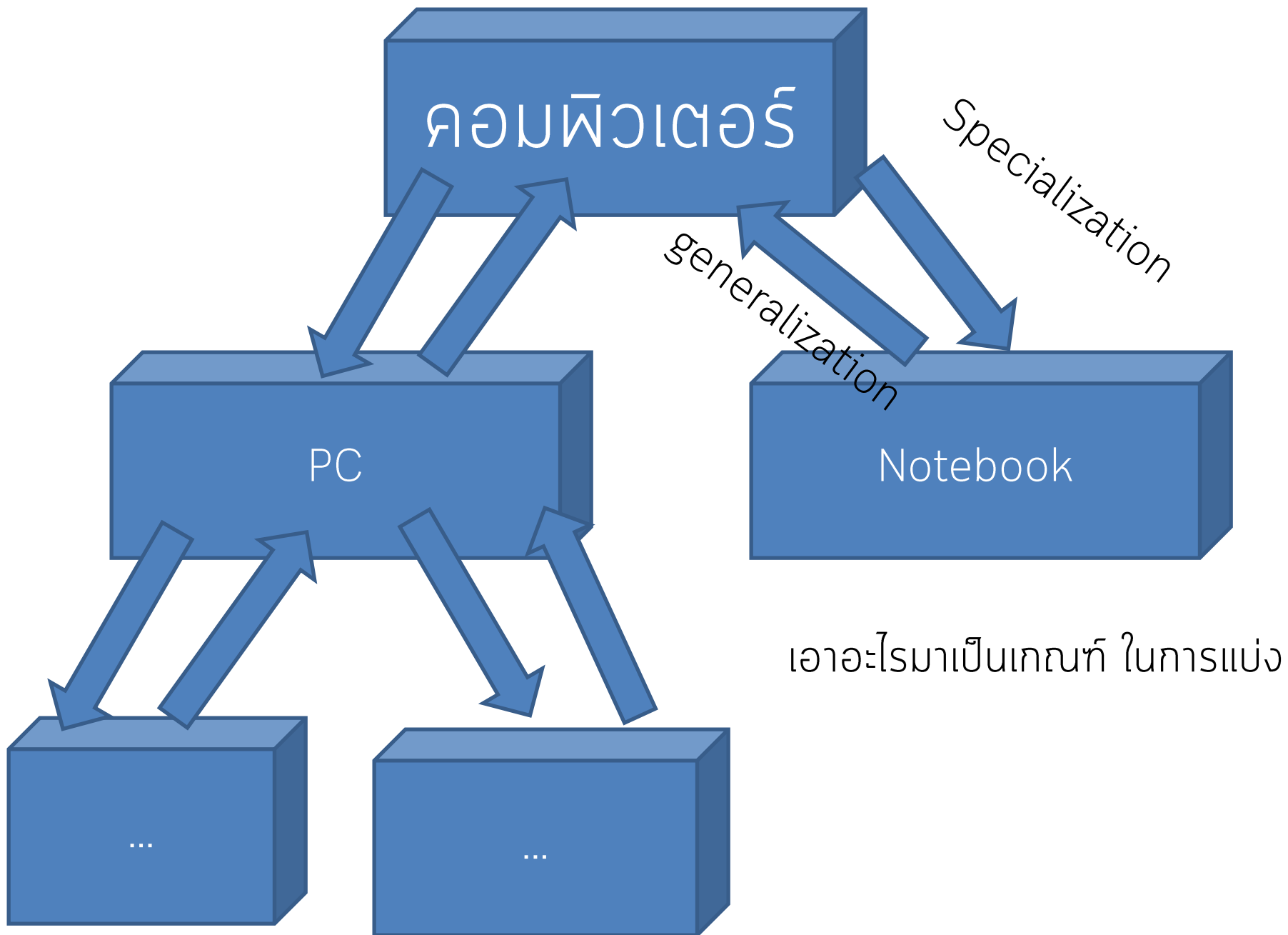
Generalization Abstraction คือกระบวนการในการนำ Class ที่มีลักษณะเหมือนหรือคล้ายกันหรือมีคุณสมบัติอย่างไรอย่างหนึ่งร่วมกัน (General) มาจัดหมวดหมู่ไว้เป็น Class เดียวกัน ซึ่งกระบวนการย้อนกลับของ Generalization Abstraction เรียกว่า Specialization คือการตอบคำถามว่าใน Class หนึ่ง ๆ นั้นสามารถจำแนกเป็น Class อะไรได้บ้าง

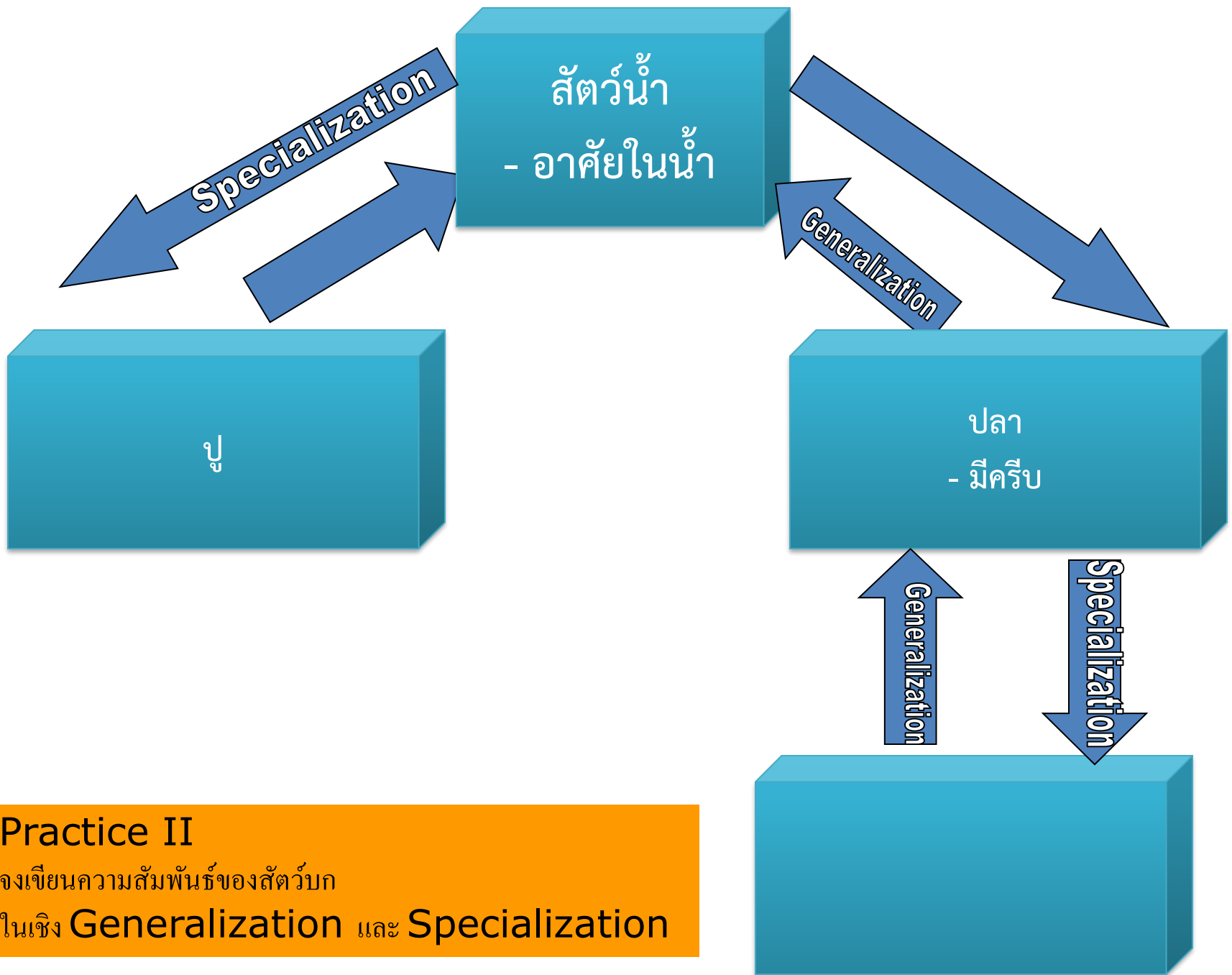












## Practice II

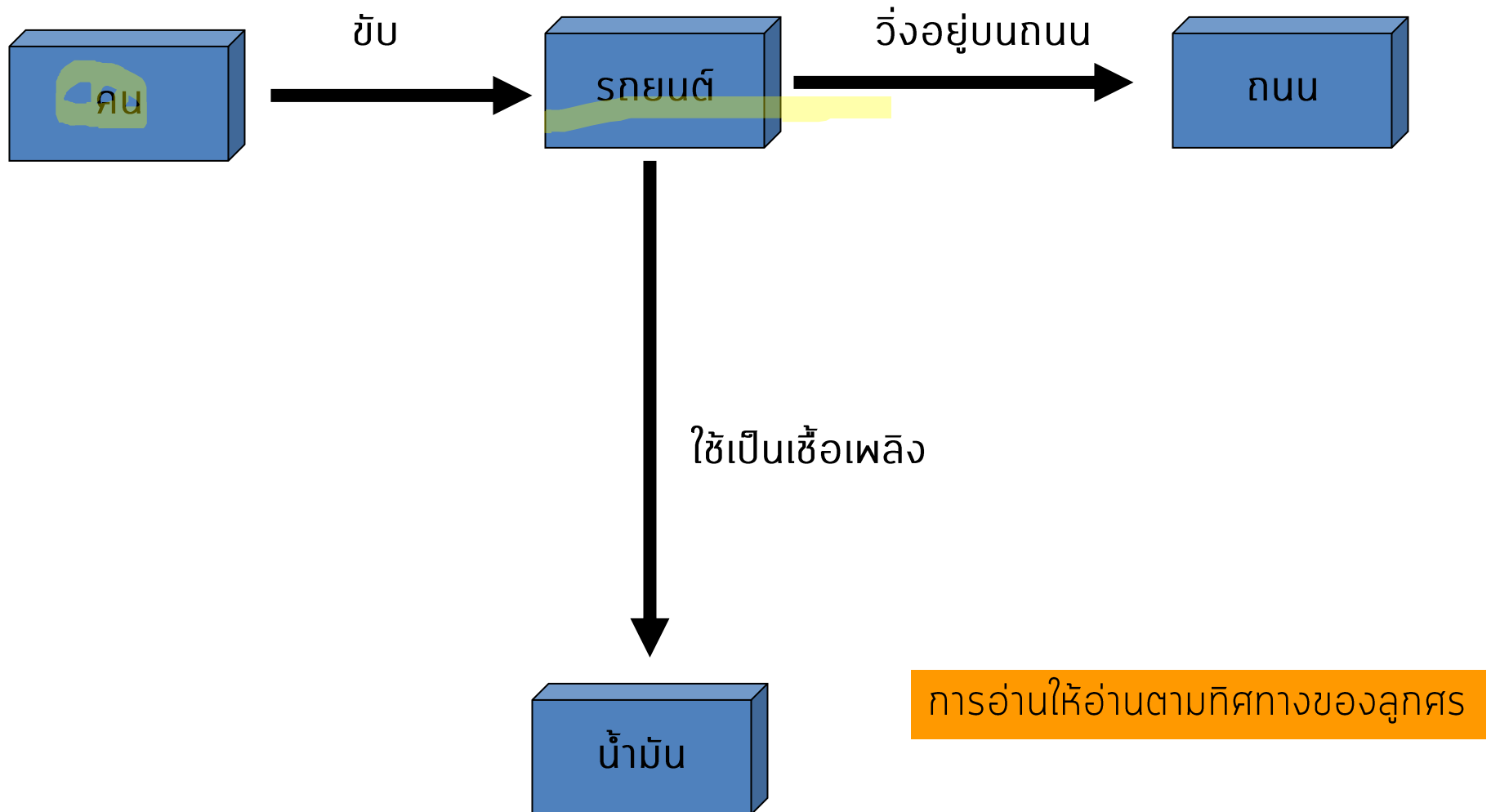
จงเขียนความสัมพันธ์ของสัตว์บก

ในเชิง **Generalization** และ **Specialization**

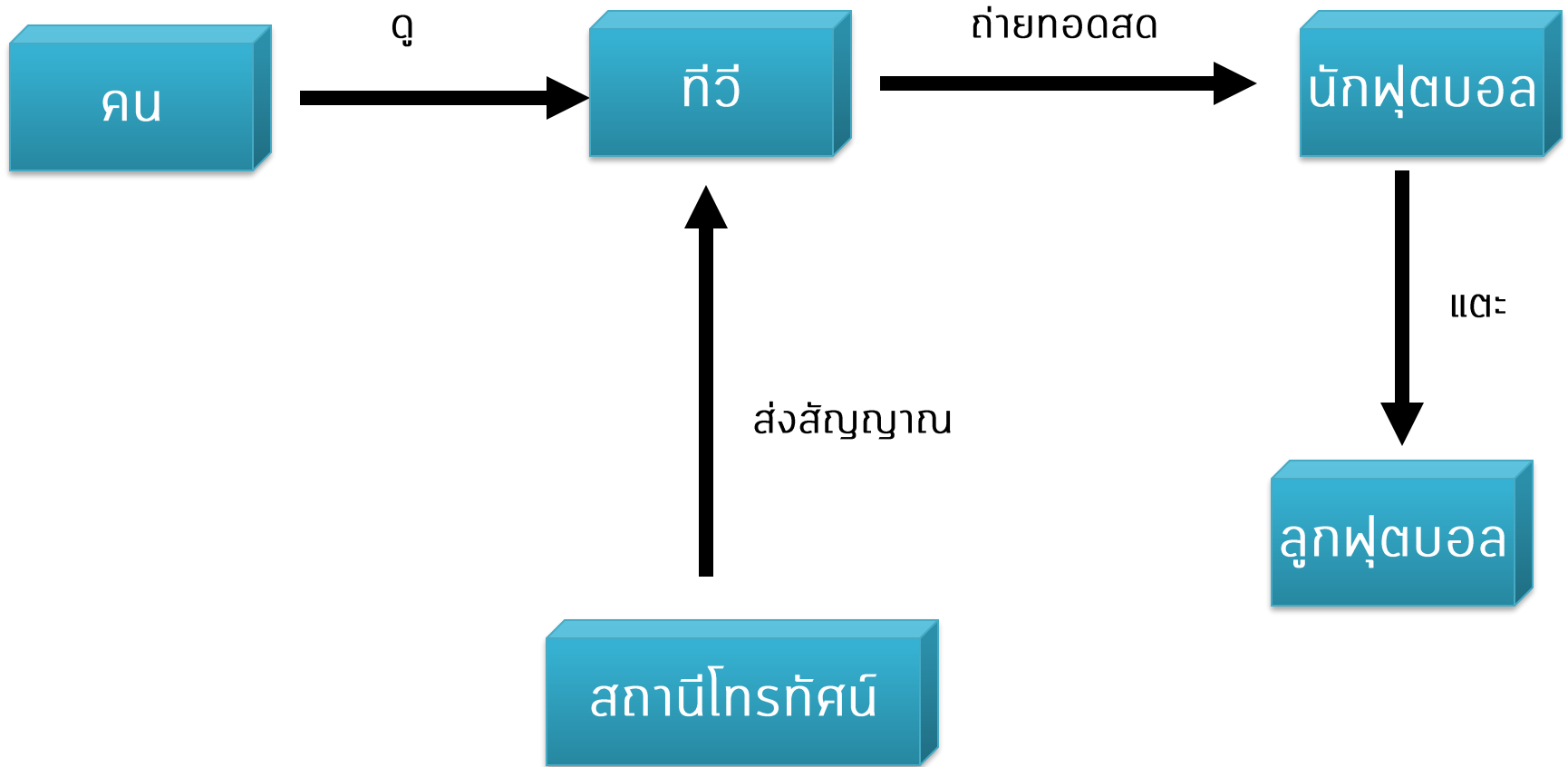
## 4. Association Abstraction

- **Association Abstraction** คือกระบวนการในการสร้างความสัมพันธ์ระหว่าง Class ต่าง ๆ ใน Problem Domain ที่เราสนใจ ความสัมพันธ์ดังกล่าวคือความสัมพันธ์ที่เราไม่สามารถอธิบายได้ด้วย Aggregation , Generalization
- เป็นการอธิบายความสัมพันธ์ในเชิงกิจกรรม เช่น การใช้งาน การขับเคลื่อน การบริโภคน เป็นต้น หรือความสัมพันธ์ ในการเป็นเจ้าของ หรือ การผลิต การให้กำเนิด

# 4. Association Abstraction



## 4. Association Abstraction



การอ่านให้อ่านตามทิศทางของลูกศร

# ตู้เย็น

Attribute 2 คะแนน

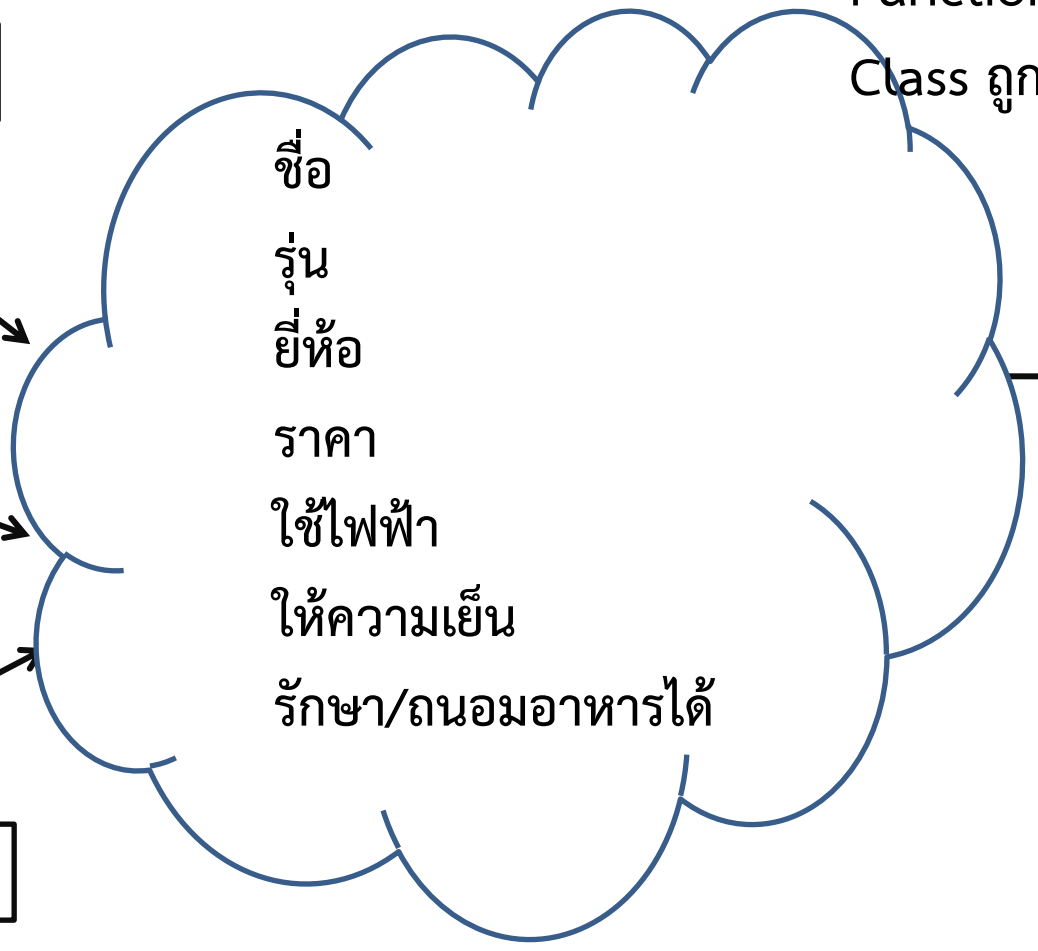
Function 2 คะแนน

Class ถูก 1 คะแนน

Mitsubishi

LG

samsung



ตู้เย็น



# แบบฝึกหัดท้ายบทที่ 2

---

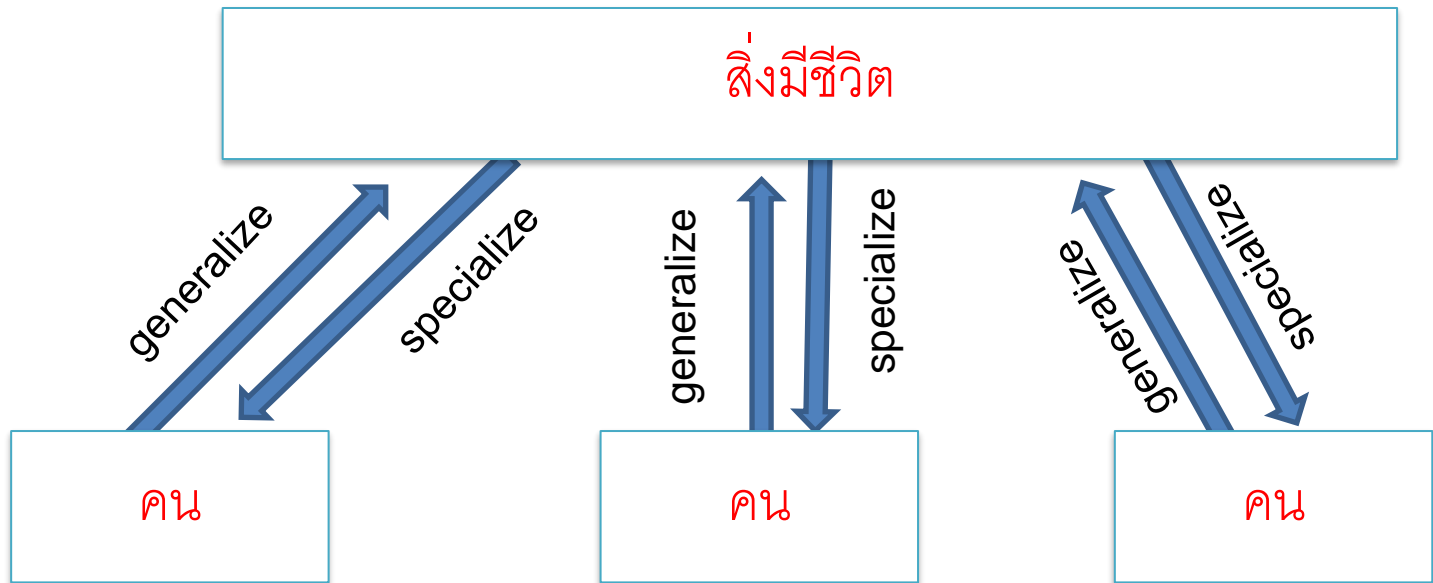
- จงให้ Concept ของสิ่งต่างๆ ต่อไปนี้
  1. Notebook
  2. โทรศัพท์มือถือ
  3. โทรทัศน์
  4. คน
  5. โรงพยาบาล
  6. ตู้เย็น
  7. ธนาคาร
  8. ตู้ ATM
  9. ห้องปฏิบัติการคอมพิวเตอร์
  10. มหาวิทยาลัย

**เลือกทำ 5 ข้อย่อย**

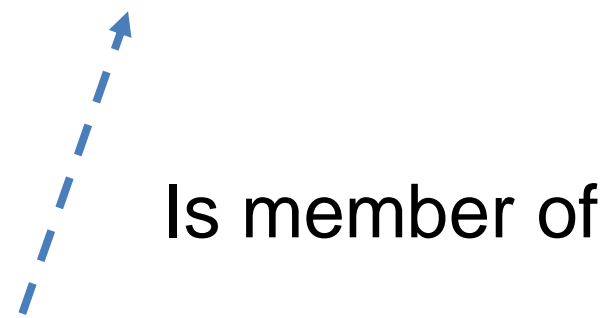
# จงบอกว่า สิ่งต่อไปนี้ อธิบายด้วย abstraction ใด

1. สิ่งมีชีวิต แบ่งออกเป็น คน สัตว์ พืช **Generalization**
2. แมวสีแดง แมวสีขาว แมวสีดำ **classification**
3. คนนั่งเครื่องบิน **association**
4. นก แบ่งออกเป็น นกเค้าแมว นกเป็ดน้ำ นกแก้ว (Generalization Abstraction)
5. โรงพยาบาล ประกอบไปด้วย หมอ พยาบาล คนไข้  
ห้องผ่าตัด ห้องคลอด (Aggregation Abstraction)
6. อาจารย์ สอน หนังสือวิชา วิเคราะห์และออกแบบระบบเชิงวัตถุ (Association Abstraction)

1. สิ่งมีชีวิต แบ่งออกเป็น คน สัตว์ พืช **Generalization**



**association**

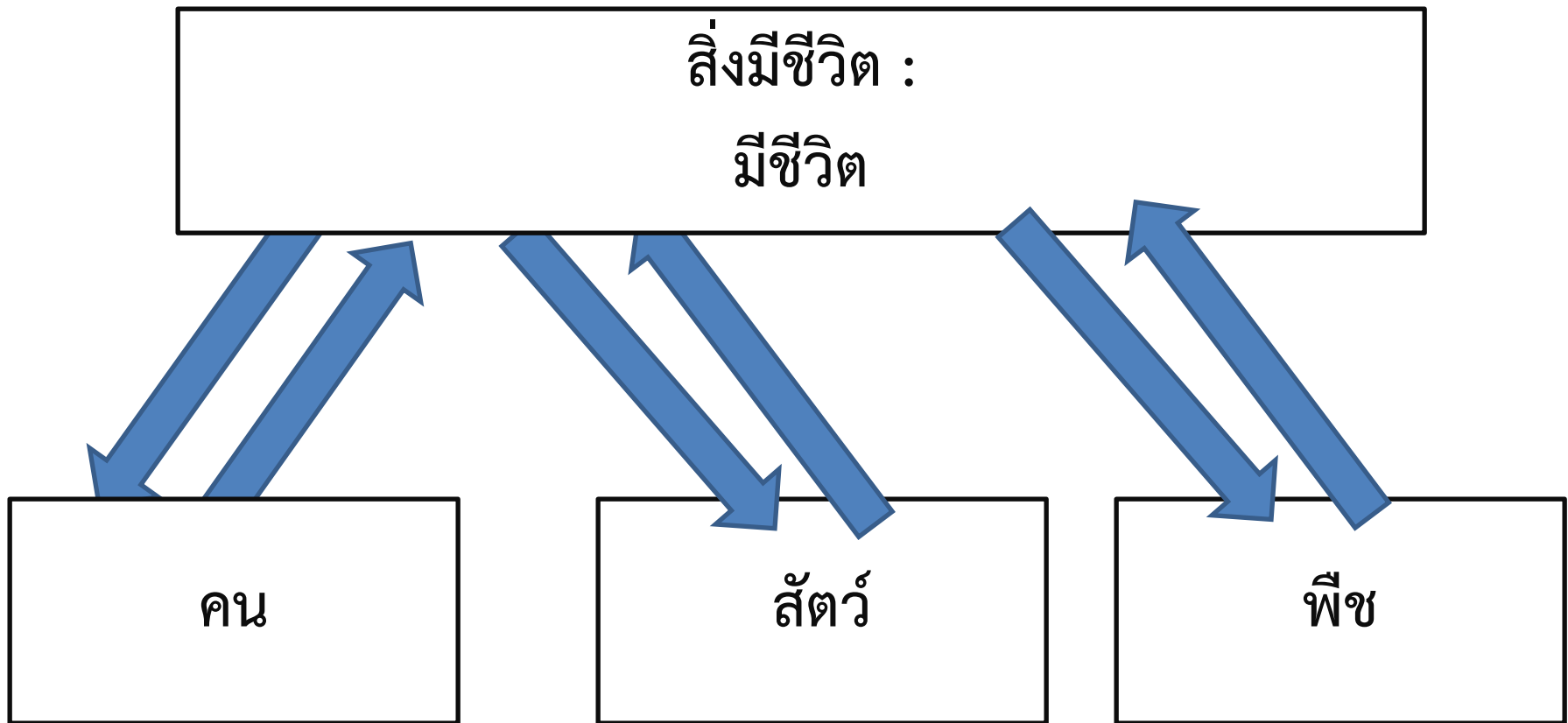


**classification**

- สิ่งมีชีวิต แบ่งออกเป็น คน สัตว์ พืช

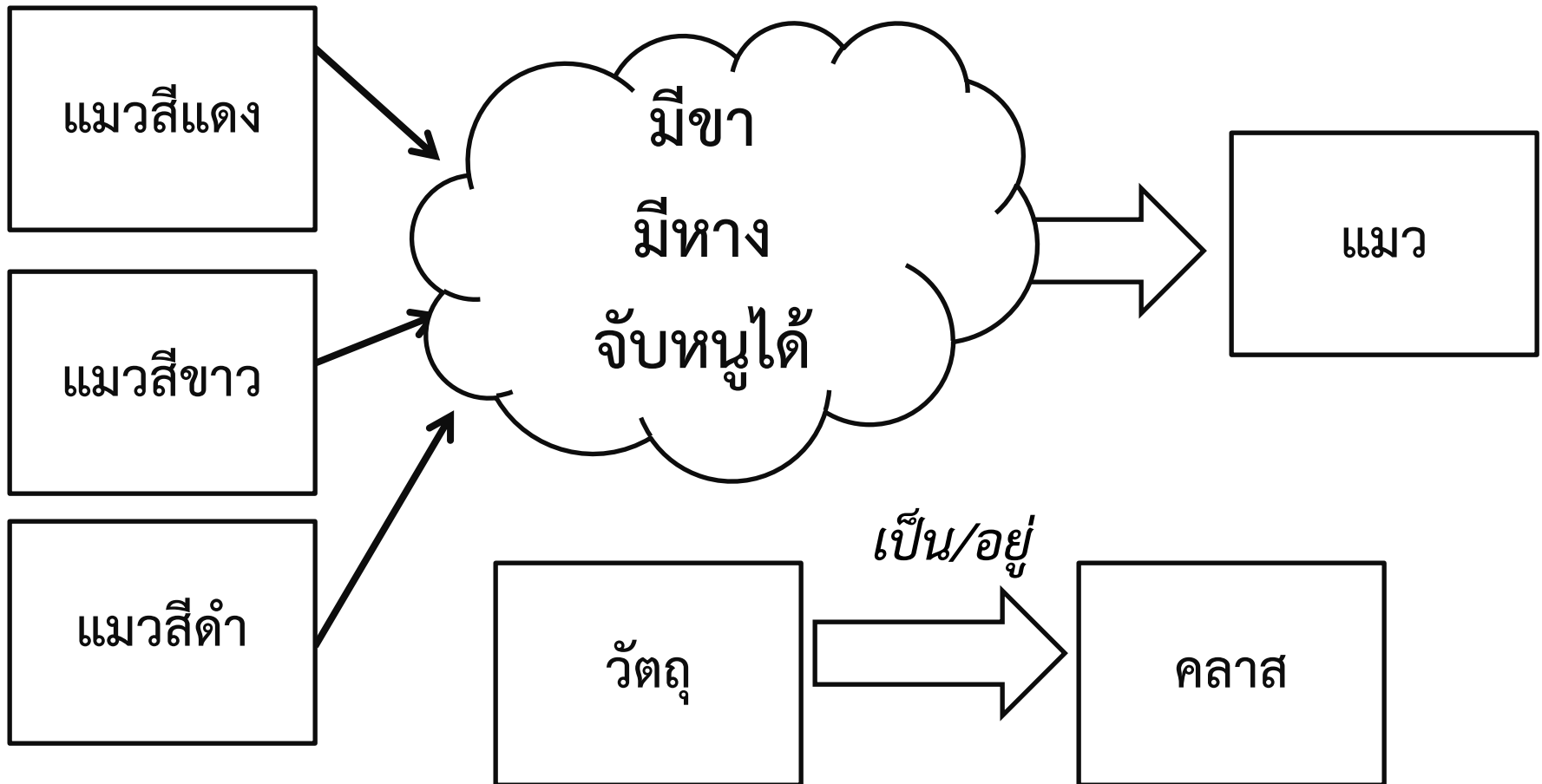
- Abstraction : เราควรเลือกใช้

## Generalization/Specialization



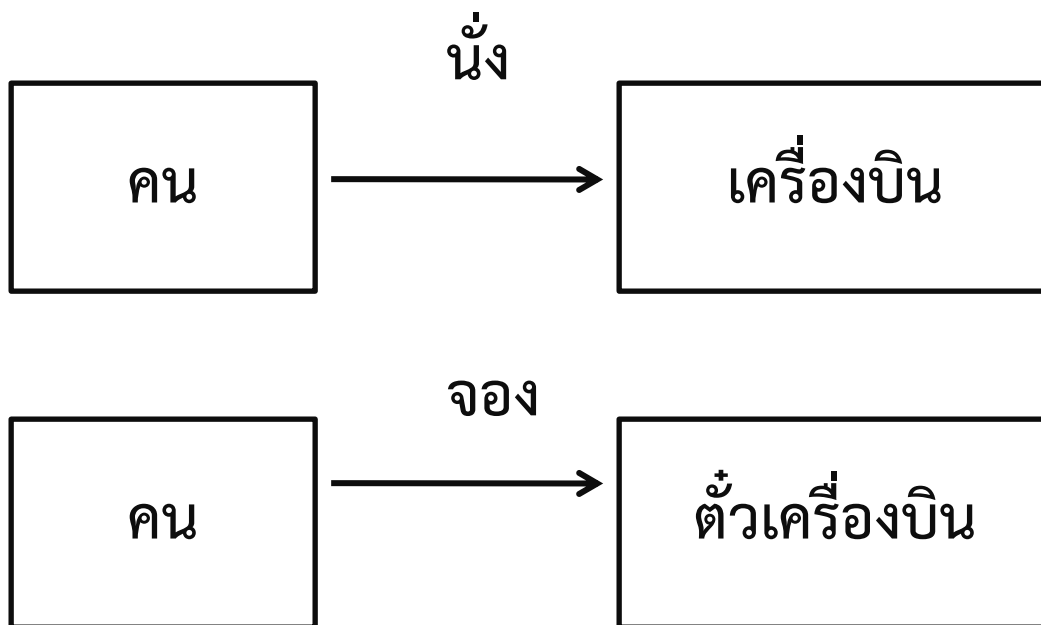
# แมวสีแดง แมวสีขาว แมวสีดำ

- Classification Abstraction



# คนนั่งเครื่องบิน

- Associations Abstraction



# โรงพยาบาล ประกอบไปด้วย หมอ พยาบาล คนใช้ ห้องผ่าตัด ห้องคลอด (Aggregation Abstraction)

- Aggregation Abstraction

