



www.siam2dev.com

## วิชา การเขียนโปรแกรมเชิงวัตถุ (4122309)

### บทที่ 2 : แนวคิดเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ ( Object Oriented Programming Concept)



ผู้ช่วยศาสตราจารย์ ดร. นัฐพงศ์ ส่องเนียม

Asst. Dr. Nattapong Songneam

เว็บไซต์ :: <http://www.siam2dev.com>

อีเมล :: [siam2dev@gmail.com](mailto:siam2dev@gmail.com)

สาขาวิชาวิทยาการคอมพิวเตอร์

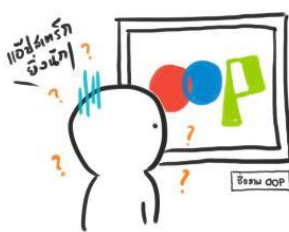
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏพระนคร

Last Update : 17/07/2566



# บทที่ 2 : แนวคิดเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุ ( Object Oriented Programming Concept)

## Agenda



1. แนวคิดการเขียนโปรแกรมเชิงวัตถุ

2. โปรแกรมภาษาเชิงวัตถุ

3. Object Orientation

4. class & Object

5. Encapsulation

6. Polymorphism

7. Inheritance



---

# แนวคิดของการเขียนโปรแกรมเชิงวัตถุ Object Orientation Concept





# การพัฒนาโปรแกรม ออกเป็น 2 ประเภท

- การพัฒนาโปรแกรมแบบโครงสร้าง (Structural Programming / Traditional Programming)

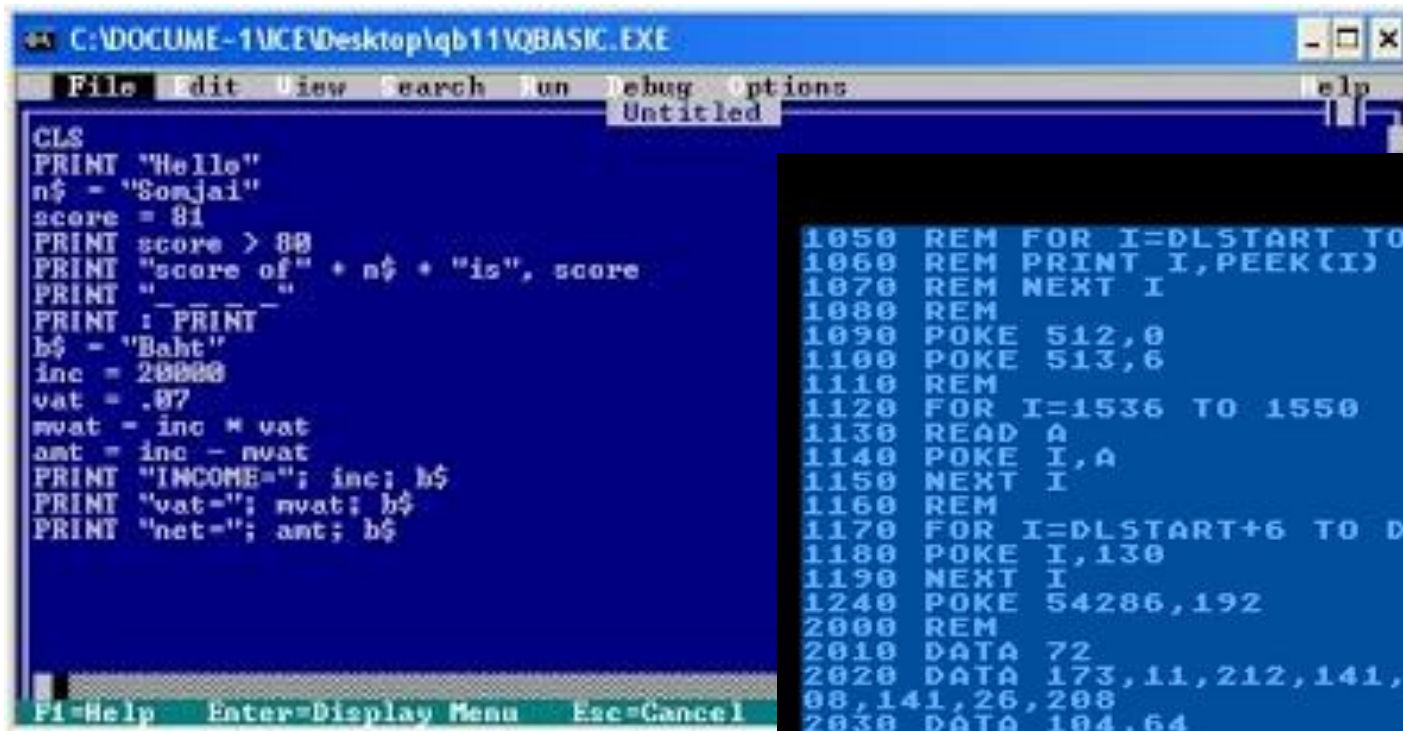
-----1960-1970 -----

c, pascal, basic, gwbasic, cobol-----

การพัฒนาโปรแกรมแบบเชิงวัตถุ Object Oriented Programming : OOP

# ภาษา Basic (ไม่ใช่ VB)

- GOTO ไปยังบรรทัดคำสั่งต่าง ๆ
- โปรแกรมมีการกระโดดไปยังบรรทัดต่าง ๆ ทำให้มองว่าเป็นภาษาโปรแกรมแบบไม่มีโครงสร้างที่ชัดเจน



```
C:\DOCUME~1\ICE\Desktop\qb11\QBASIC.EXE
File Edit View Search Run Debug Options Help
Untitled
CLS
PRINT "Hello"
n$ = "Somjai"
score = 81
PRINT score > 88
PRINT "score of" + n$ + "is", score
PRINT " "
PRINT : PRINT
b$ = "Baht"
inc = 20000
vat = .07
nvat = inc * vat
amt = inc - nvat
PRINT "INCOME="; inc; b$
PRINT "vat="; nvat; b$
PRINT "net="; amt; b$

1050 REM FOR I=DLSTART TO DLEND
1060 REM PRINT I,PEEK(I)
1070 REM NEXT I
1080 REM
1090 POKE 512,0
1100 POKE 513,6
1110 REM
1120 FOR I=1536 TO 1550
1130 READ A
1140 POKE I,A
1150 NEXT I
1160 REM
1170 FOR I=DLSTART+6 TO DLSTART+28
1180 POKE I,130
1190 NEXT I
1240 POKE 54286,192
2000 REM
2010 DATA 72
2020 DATA 173,11,212,141,10,212,141,24,2
08,141,26,208
2030 DATA 104,64
```

# ภาษาโครงสร้าง Structural Programming

- C
- Pascal

- มีรูปแบบตายตัว
- มีโครงสร้างที่ชัดเจน
- การเริ่มต้นสิ้นสุด

```
File Edit Run Compile Options Debug Break/watch
Line 7 Col 5 Insert Indent Unindent * C:NONAME.PAS
PROGRAM SAMPLE1;
Begin
  Write('Hello ! How are You?');
  Write('My Name Is Rungrat');
  Write('Nangrong Technical College');
End...
```

ไม่เห็นหน้าตาผลลัพธ์ is not WYSIWYG :  
What You See It What You Get

# ก่อนหน้า 1960 เป็นภาษาเชิงโครงสร้าง c, pascal Structure Programming Language

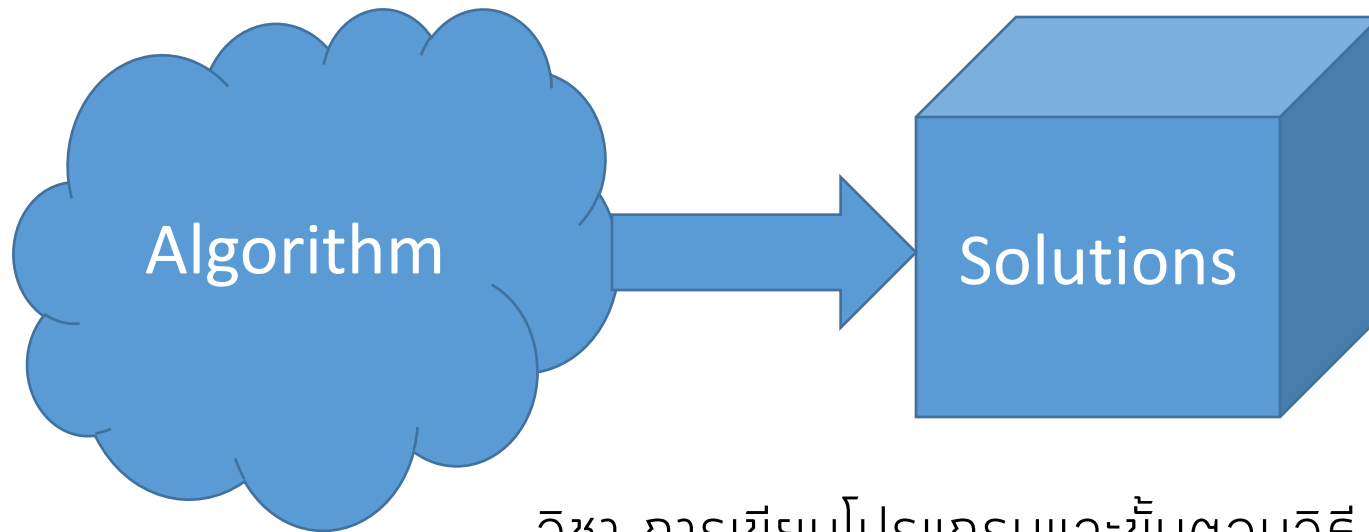
- โดย มองปัญหา และแก้ปัญหาโดยใช้ขั้นตอนวิธี หรือ algorithm
  - Algorithm / **อัลกอริทึม** / ขั้นตอนวิธี
  - คือการแก้ไขปัญหามาแบบเป็นขั้นเป็นตอน
- แบบลำดับ (Sequential) จากบรรทัดแรกไปยังบรรทัดสุดท้าย
- รูปแบบการทำงานที่ชัดเจน

ดังนั้นผู้ที่เขียนโปรแกรมได้นั้นจะต้องมีความรู้ 2 เรื่องเป็นอย่างน้อย

Algorithm + Syntax

# Structural Programming

- หาวิธีการแก้ปัญห โดยใช้ Algorithm



วิชา การเขียนโปรแกรมและขั้นตอนวิธี (Algorithm)

Mathematic + English

Logic: ตระการ selection , loop , module

# Algorithm

- การบอกวิธีการมาเป็นข้อ ๆ
- สั้น
- กระจ่างชัด
- ได้ใจความ
- สามารถทำงานได้ทันที

Algorithm + Syntax

Analysis

Design

จะต้องรู้ว่าปัญหา นั้นจะต้องแก้ไขอย่างไร

Flowchart

Pseudo-Code

StructureChart

Mathematic + English

Logic: assign selection , loop , module

รูปแบบภาษา ที่ใช้พัฒนา

ภาษาเชิงโครงสร้าง

ภาษาเชิงวัตถุ

Object-Oriented

# จงหาอายุเฉลี่ยของ นศ. วิทยาลัยคอม 60-1 ?

• Algorithm(PSEUDO CODE) + Flowchart

- สิ่งที่ต้องการ ... โจทย์ต้องการหาอายุเฉลี่ยของนศ. วิทยาลัยคอม 60-1
- รูปแบบผลลัพธ์ หน้าจอ ที่ต้องการให้เป็น

อายุเฉลี่ยของ นศ. วิทยาลัยคอม 60- 1 เท่ากับ xx ปี

- ข้อมูลนำเข้า
  - อายุของนักศึกษาแต่ละคน
- ตัวแปรที่ใช้
  - กำหนดตัวแปรที่จำเป็น
  - ตัวแปร INPUT .... Age ... แทน อายุแต่ละคน
  - ตัวแปร OUTPUT .... AVG แทน ค่าเฉลี่ยของอายุ นศ. ทั้งห้อง (Average)
  - ตัวแปรที่ใช้ PROCESS ... Total แทน อายุรวมของ นศ. ทุกคน

ออกแบบ Mockup wire flame



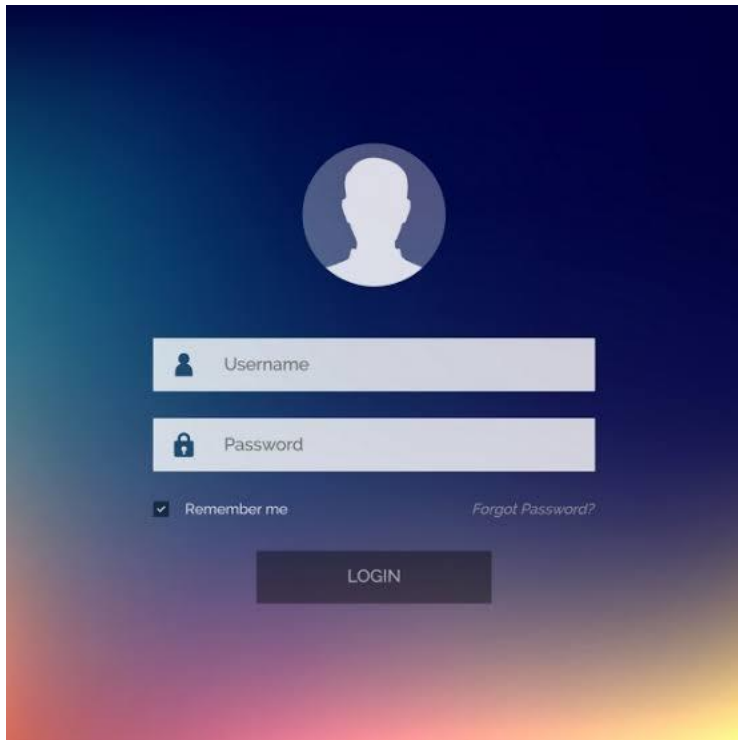
HCI: Human Computer Interaction

Analysis

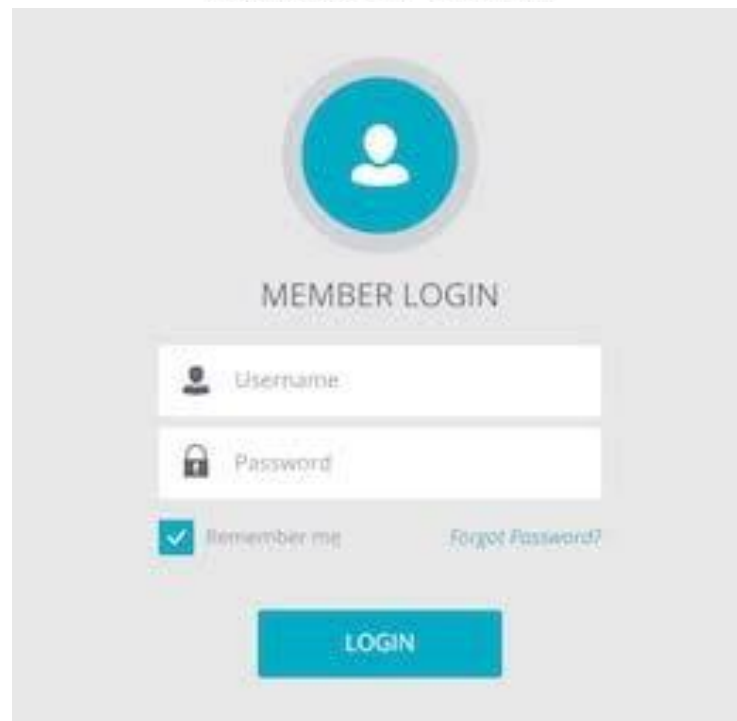
1. สิ่งที่ต้องการ
2. รูปแบบผลลัพธ์
3. ข้อมูลนำเข้า
4. ตัวแปรที่ใช้
5. วิธีการประมวลผล



# Login Design



shutterstock.com • 501077941



ประวัติพนักงาน Employee

ทะเบียน Record: 1/1

51001 นมิศร สุภอัครโสภณ, ประเภท S(พนักงานรายเดือน)

ทั่วไป สถานะ เปลี่ยนคืน ค่าคงช่อก่อน และวันสะสม รายการรับ/หักคงที่ รายการรับ/หักเปลี่ยนแปลง

รหัสพนักงาน 51001

รหัสรอง

ตำแหน่ง

ประเภท

แผนก/ฝ่าย

เลือกภาพ

ชื่อ-สกุล นมิศร สุภอัครโสภณ

ที่อยู่

จังหวัด

รหัสไปรษณีย์

โทรศัพท์

สถานะ S - ไลด

ประเภท S - พนักงานรายเดือน

รหัสธนาคาร S015 - ธนาคารนครหลวงไทย จำกัด

บัญชีธนาคาร#

ผู้เบิกภาษี#

บัตรประชาชน#

สำรวจรายการโอนข้อมูลรับ และหักคงที่

ต้องการสร้าง...  
รายการโอนข้อมูลรับหักคงที่ และเปลี่ยนแปลง  
สำหรับพนักงานคนนี้หรือไม่?

Yes No

บันทึกข้อมูลพนักงาน : 1330050 นาย ธีระพงศ์ ปรสภักดิ์

New Save Save and New Save and Close Print More Action Close

บันทึกข้อมูลพนักงาน

Details

Information Other menu

ค่าคงช่อก่อนที่ ค่าคงช่อก่อน เบื้องกลาง ค่าคงช่อก่อน คนหักหัก ประกันชีวิต การหักเกษม กลุ่มผู้ถือ / สิทธิคนหัก ที่ผู้ถือเงิน

Administrator Note Attachment

General สถานภาพส่วนตัว การจ้างงาน การชำระเงิน เงินได้ก่อนและหลัง ค่าได้-หักหัก ประกันสุขภาพ

รหัสพนักงาน 1330050  สถานภาพผู้ถือ

หมายเลขบัตร 1330050 เลขที่ใบสมัคร

หมายเลขบัตรชั่วคราว

ชื่อพนักงาน นาย ธีระพงษ์ ปรสภักดิ์

ชื่อพนักงาน (Eng) Mr. Teerapong Onpakdee

ชื่อเล่น

วันเกิด 01/01/2516 อายุ 37 ปี

เพศ ชาย สถานะโสด

ส่วนสูง 170.00 ซม. ประเภท ไทย


น้ำหนัก 60.00 กก. มีสูบบุหรี่ กรุงเทพมหานคร

หมู่เลือด AB

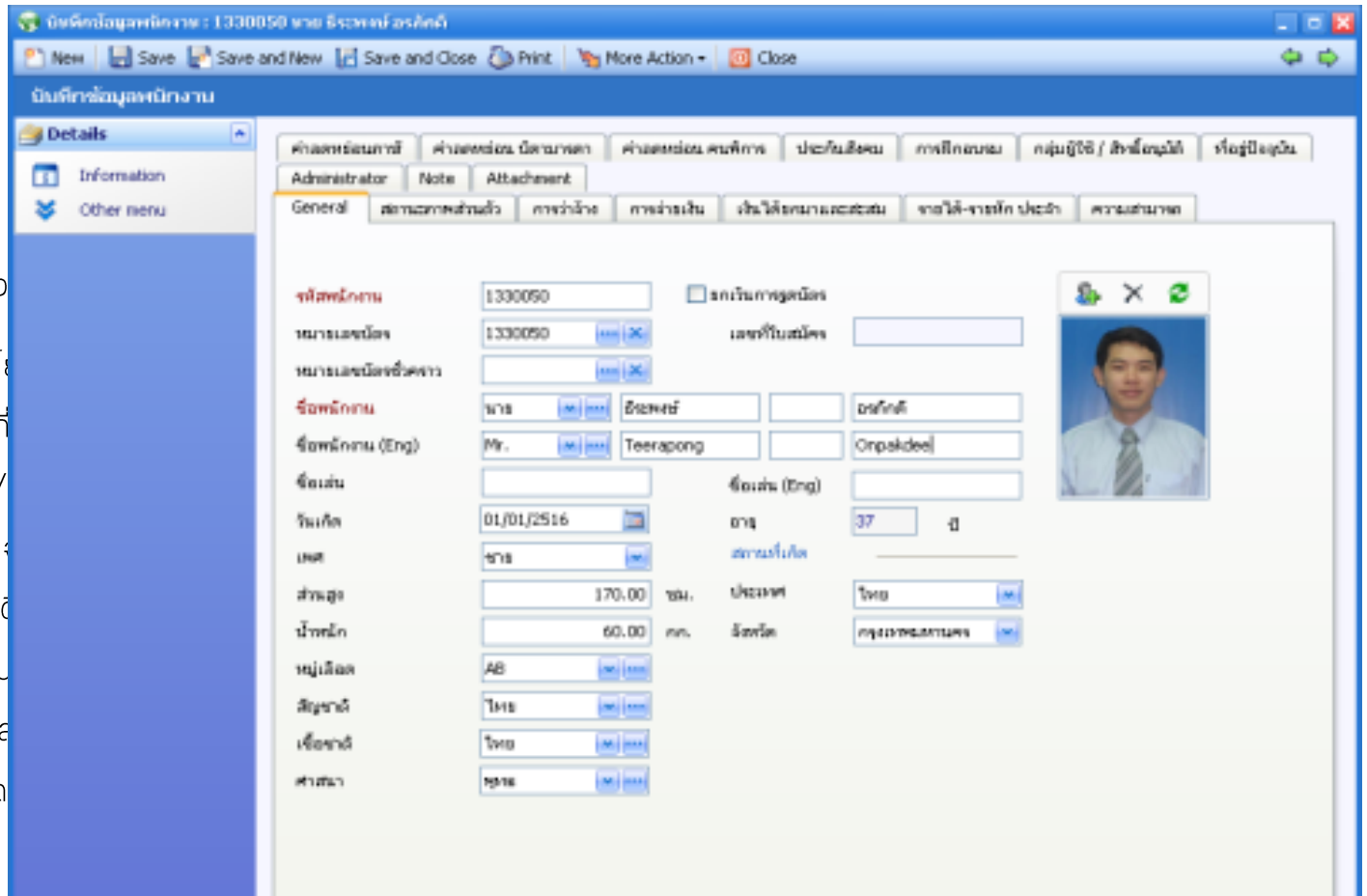
สัญชาติ ไทย

เชื้อชาติ ไทย

ศาสนา พุทธ



- ใช้สำหรับบันทึกประวัติของพนักงานในองค์กร เพื่อเก็บข้อมูลต่างๆ ของพนักงาน เช่น สถานะภาพส่วนตัว การว่าจ้าง การจ่ายเงิน รายได้ - รายหักประจำ ความสามารถ ค่าลดหย่อนภาษี ฯลฯ **Benefit :**
- สามารถเก็บประวัติของพนักงานได้โดยละเอียดซึ่งมี Functional ดังนี้ - ข้อมูลส่วนตัว
  - สถานะภาพส่วนตัว
  - สถานะการทำงานปัจจุบัน
  - ข้อมูลรายได้/รายหัก เงินเดือน
  - บุคคลคำประกัน
  - ประวัติการศึกษา
  - ประสบการณ์การทำงาน
  - ความสามารถ
  - โรคประจำตัว
  - ประวัติการรักษาพยาบาล
  - ข้อมูลทางครอบครัว
  - ที่อยู่อาศัย
  - ค่าลดหย่อน
  - สถานภาพการเสียภาษีของ
  - ข้อมูลประกันสังคม
  - เงินสะสมกองทุนสำรองเลี้ยงชีพ
- สามารถบันทึกรายละเอียดที่
- สามารถบันทึกการลาออก /
- สามารถแสดงข้อมูลสัญญาณ
- สามารถแสดงข้อมูลผลงาน
- สามารถแสดงข้อมูลการปรับ
- สามารถแสดงข้อมูลการถูก
- สามารถแสดงข้อมูลการขาด



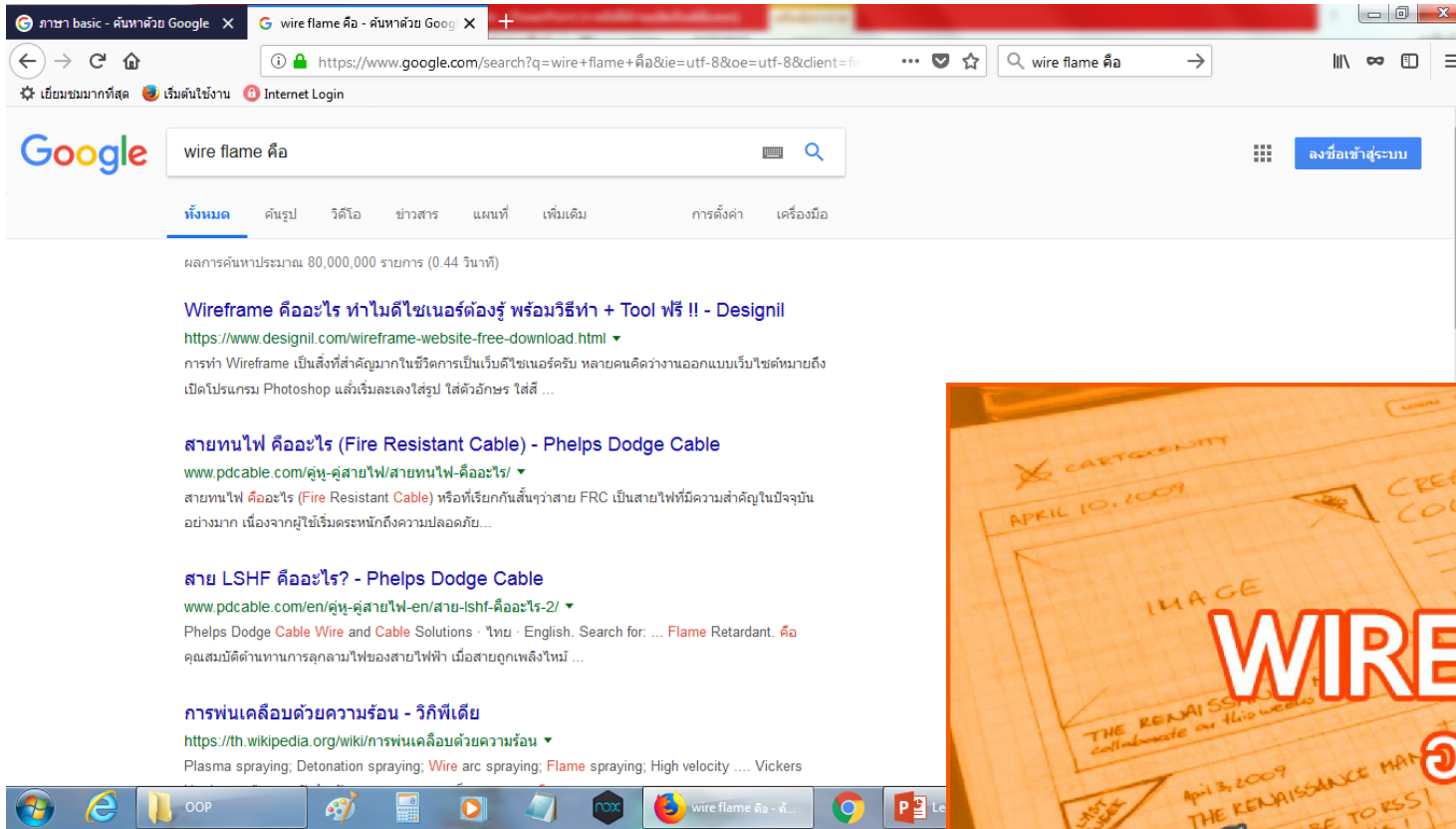
สภาพภาพส่วนตัว

ข้อมูลรายได้

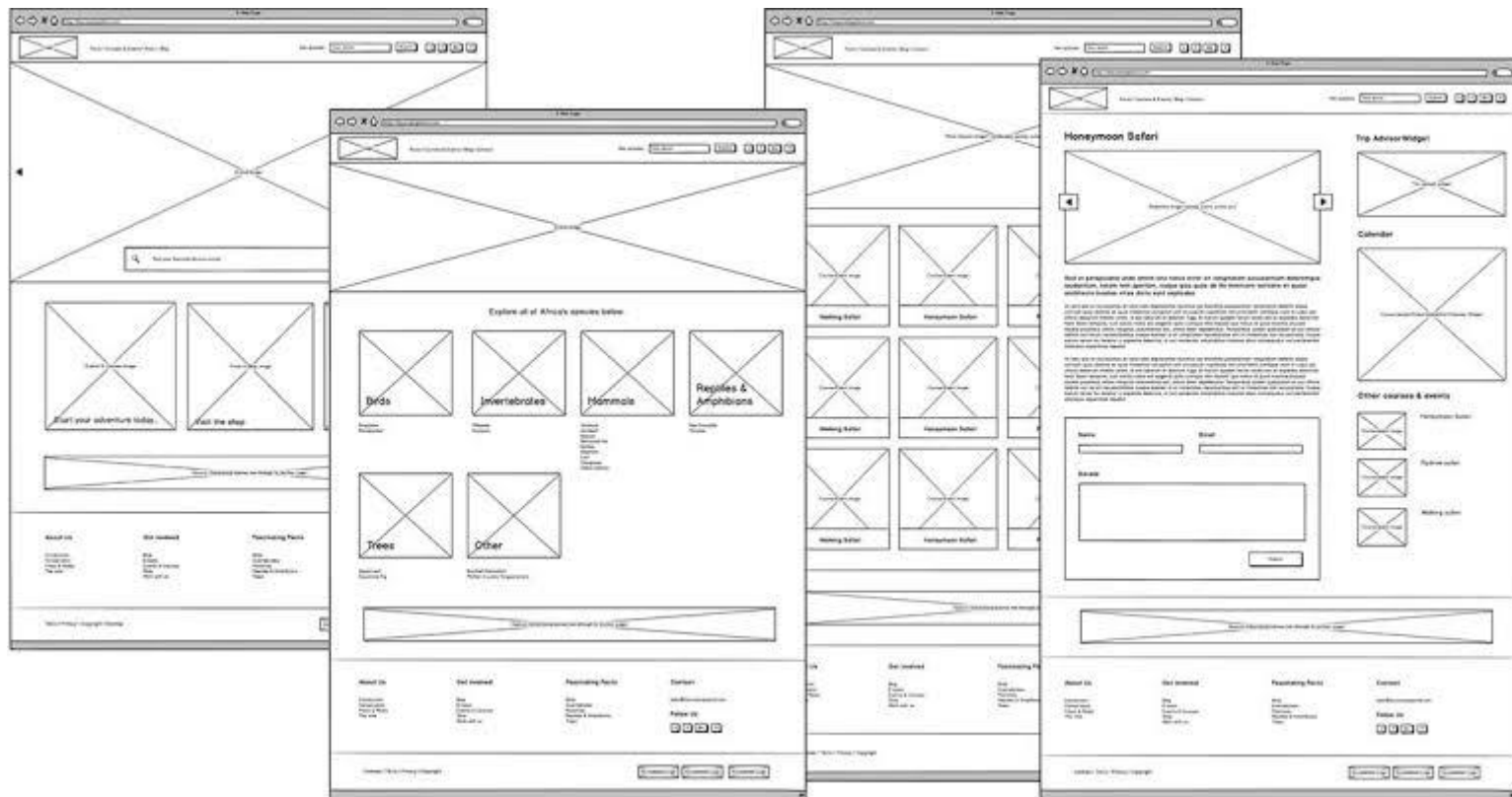
ข้อมูลรายได้

พนักงาน

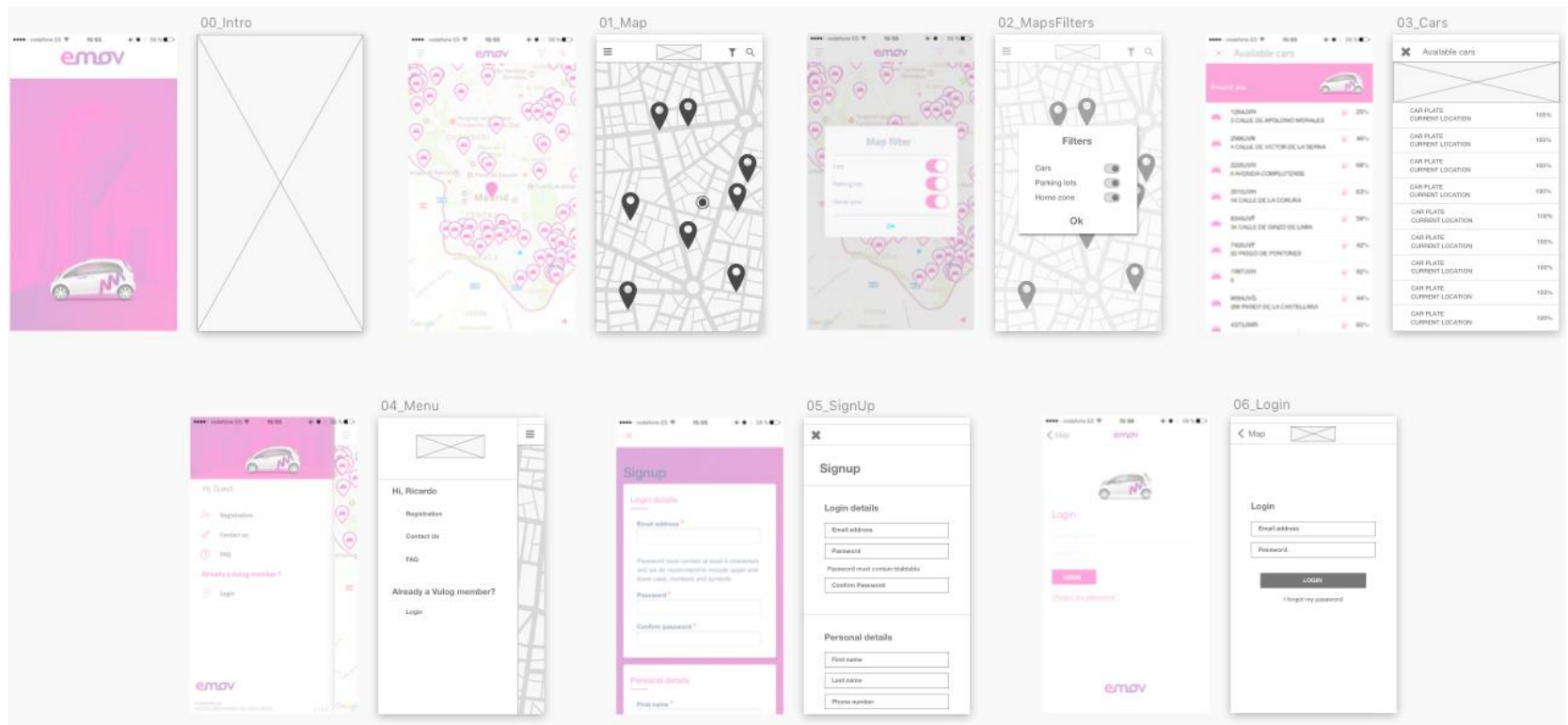
# wire flame คือ



# wireframe



# Mockup





# องศาผลการเรียนเฉลี่ยของนักศึกษา 1 ภาคการศึกษา

## Analysis

1. สิ่งที่ต้องพิจารณา
2. รูปแบบผลลัพธ์
3. ข้อมูลนำเข้า
4. ตัวแปรที่ใช้
5. วิธีการประมวลผล

- สิ่งที่ต้องพิจารณา คือ ผลการเรียนเฉลี่ยของนักศึกษา 1 ภาคการศึกษา
- .....รูปแบบผลลัพธ์

### ผลการเรียนภาคการศึกษา ..X../XXXX

ชื่อ-นามสกุล ..... สาขา.....

| รหัสวิชา | ชื่อวิชา | หน่วยกิต | เกรด |
|----------|----------|----------|------|
| XXXXXX   | XXXXXXXX | x(x-x)   | x.xx |
| XXXXXX   | XXXXXXXX | x(x-x)   | x.xx |
| XXXXXX   | XXXXXXXX | x(x-x)   | x.xx |
|          |          | GPA      | x.Xx |

### ข้อมูลนำเข้า

- รายวิชา รหัส ชื่อ
- เกรด
- หน่วยกิต
- ชื่อ นศ. รหัส สาขา
- ภาคการศึกษา

สิ่งที่ไม่ต้องนำเข้า .....อะไรที่เป็นค่าตายตัว คงตัว คงที่ สากล พื้นฐาน



สิ่งที่ไม่ต้องนำเข้า .....อะไรที่เป็นค่าตายตัว คงตัว คงที่ สากล พื้นฐาน

- ลำดับ 1 2 3 4.....
- ค่า PI , VAT , .....
- ค่า c 1/2

# ตัวแปรที่ใช้

- ตัวแปรที่ใช้
  - กำหนดตัวแปรที่จำเป็น
    - ตัวแปร INPUT .... Age ... แทน อายุแต่ละคน
    - ตัวแปร OUTPUT .... AVG แทน ค่าเฉลี่ยของอายุนศ. ทั้งหมด (Average)
    - ตัวแปรที่ใช้ PROCESS ... Total แทน อายุรวมของ นศ. ทุกคน
- Student Name แทน ชื่อนักศึกษา
- Student ID :
- GRADE
- CREDIT
- GPA
- TOTAL
- SEMESTER แทนภาคการศึกษา

# สิ่งที่ไม่ต้องนำเข้า .....OUTPUT ของโปรแกรม

- จงหาค่า พ.ท. สามเหลี่ยม
  - ฐาน
  - สูง
  - ค่า พ.ท. สามเหลี่ยม

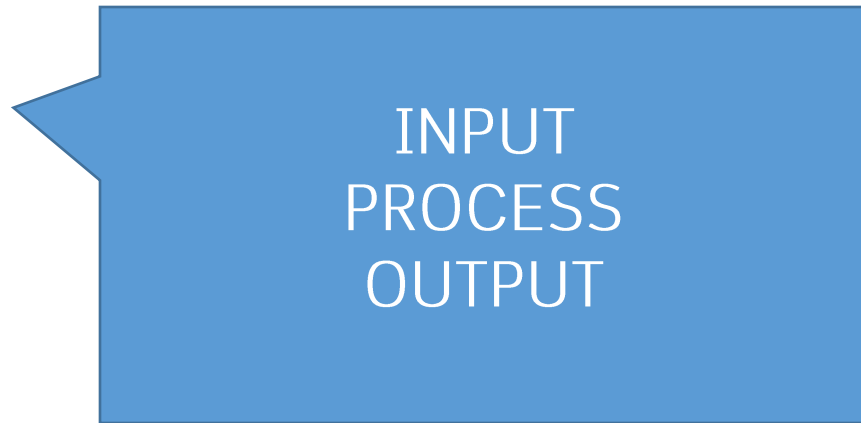
# วิธีการประมวลผล การออกแบบว่าจะทำที่ละขั้นตอน อย่างไร Algorithm

1. เริ่มต้น
2. ... ???
3. สิ้นสุด / จบ /  
เสร็จสิ้น

แยกมาเป็นข้อ ๆ ที่สามารถ  
ทำงาน 1 อย่าง  
เป็นคำสั่งขั้นพื้นฐาน ที่สามารถ  
เข้าใจได้ โดยไม่มีคำถามต่อ

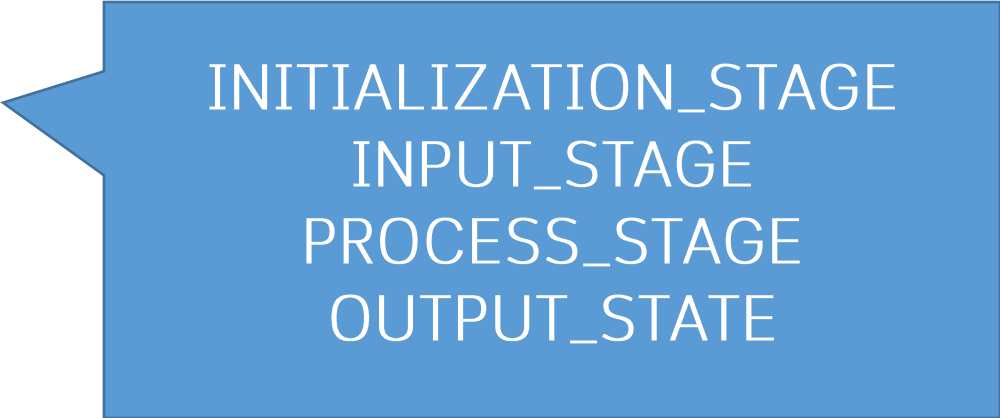
# วิธีการประมวลผล การออกแบบว่าจะทำที่ละขั้นตอน อย่างไร Algorithm

1. เริ่มต้น
2. ... ???
3. สิ้นสุด / จบ /  
เสร็จสิ้น



# วิธีการประมวลผล การออกแบบว่าจะทำที่ละขั้นตอน อย่างไร Algorithm

1. เริ่มต้น
2. ... ???
3. สิ้นสุด / จบ /  
เสร็จสิ้น



INITIALIZATION\_STAGE  
INPUT\_STAGE  
PROCESS\_STAGE  
OUTPUT\_STAGE

# วิธีการประมวลผล การออกแบบว่าจะทำที่ละขั้นตอน อย่างไร Algorithm

1. เริ่มต้น
2. ... DEFINE X as INTEGER
  - 2.1 SET X  $\leftarrow$  0
  - 2.2 SET Total  $\leftarrow$  0
  - 2.3 SET AVG  $\leftarrow$  0.0
1. สิ้นสุด / จบ / เสร็จสิ้น

INITIALIZATION\_STAGE

# ไวยากรณ์ทางภาษา ในการเขียนโปรแกรม

Algorithm + Syntax

จงหาค่าเฉลี่ย ของตัวเลข 10 จำนวน ใช้คำสั่งอะไรบ้าง รูปแบบภาษา เขียนเป็นโปรแกรมภาษาได้อย่างไร

Case sensitive

(c,c++,java,php)

None-Case sensitive

(pascal ,Delphi , vb , vb.net)

```
Program test1;  
Begin  
  
End.
```

```
include #stdio.h  
main () {  
    int x;  
    int X;  
}
```

```
Public class form1  
    private form_load(.....)  
end sub  
End class
```



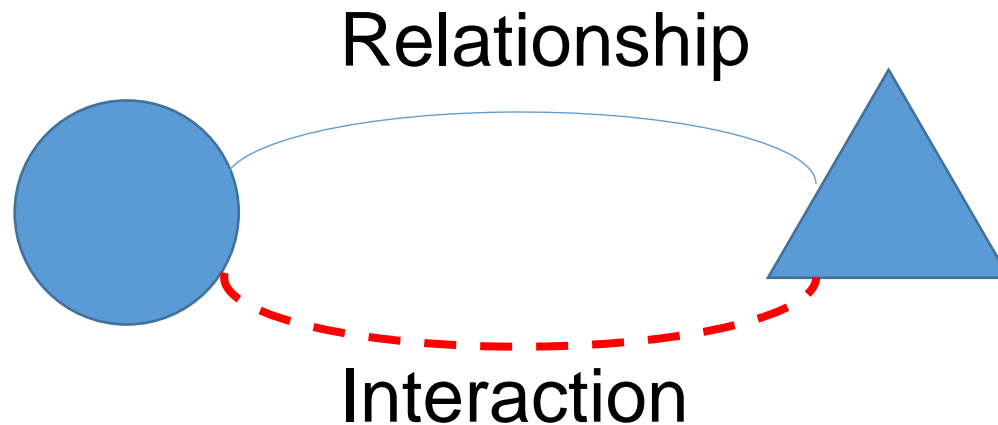
# จงหาค่าเฉลี่ย ของตัวเลข 10 จำนวน

- การวิเคราะห์ปัญหา
  - อะไรคือค่าเฉลี่ย
  - โจทย์ต้องการอะไร สิ่งที่โจทย์ต้องการ
  - ต้องรับอะไรเข้า INPUT
  - *ประมวลผลอย่างไร PROCESS ทำอย่างไร ALGORITHM*
    - *แบ่งงาน จากใหญ่ไปเล็ก*
    - *สามารถบอกเป็นข้อๆ ที่คอมพิวเตอร์ เข้าใจ และทำได้ 1 คำสั่งของภาษา*
    - *เรียงลำดับ ก่อนหลัง*
  - อะไรคือผลลัพธ์ OUTPUT

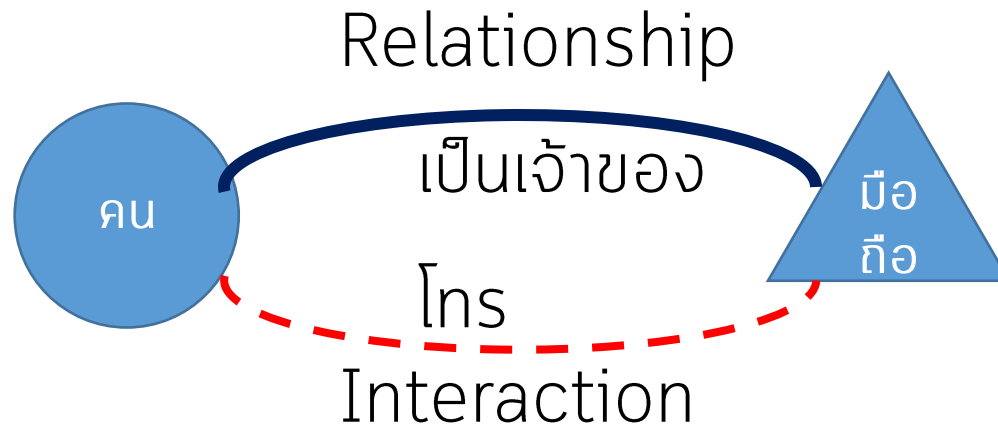
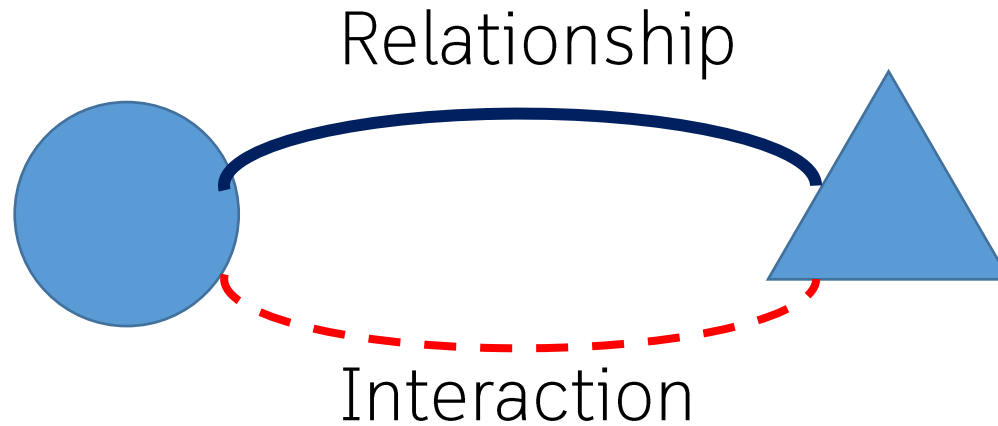


# Object Orientation

**Object Orientation** หมายถึง การใช้ Object เป็นตัวหลักเพื่อการพิจารณาความเป็นจริงต่างๆ ที่เกิดขึ้นในโลกด้วยการมองทุกๆ สิ่งในโลกให้เป็นวัตถุทั้งหมด และมองว่า กิจกรรมทั้งหมดที่เกิดขึ้นในโลกนี้เกิดจาก *ความสัมพันธ์ (Relationship)* และ *ปฏิสัมพันธ์ (Interaction)* ระหว่างวัตถุ

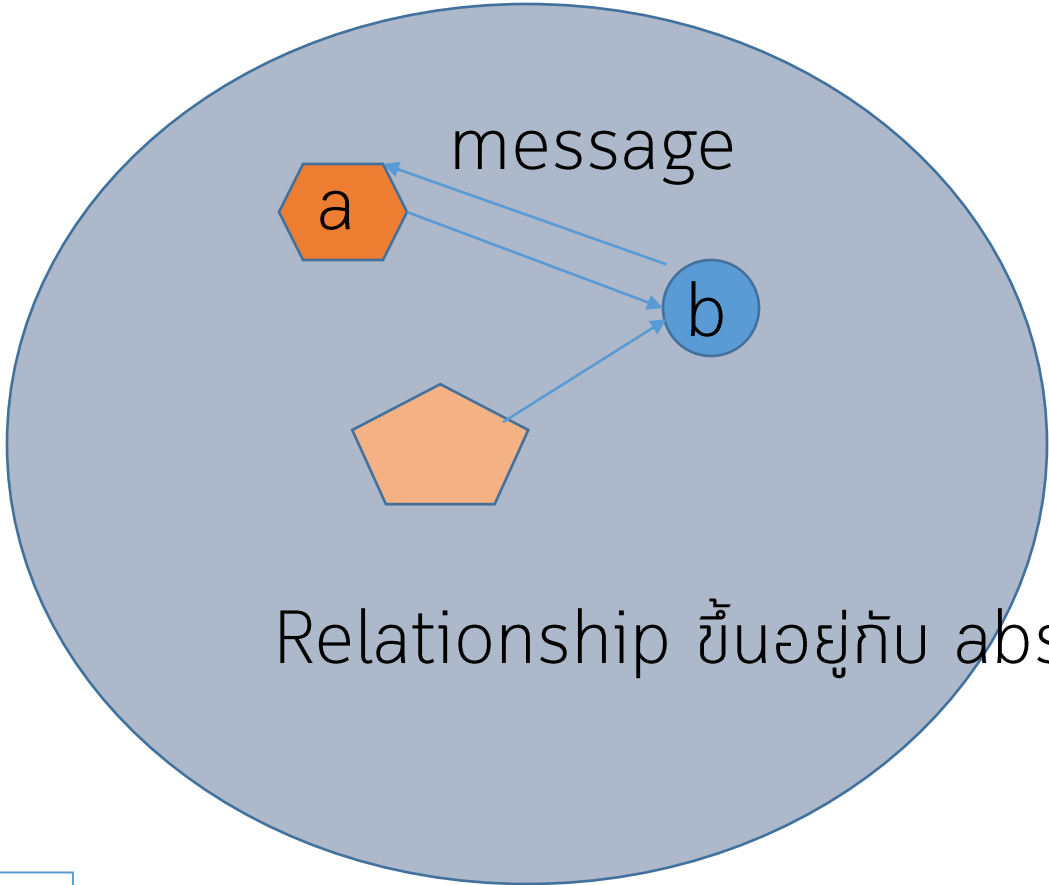


# Object Orientation



```
class B {  
  
}
```

```
class A {  
  
}
```



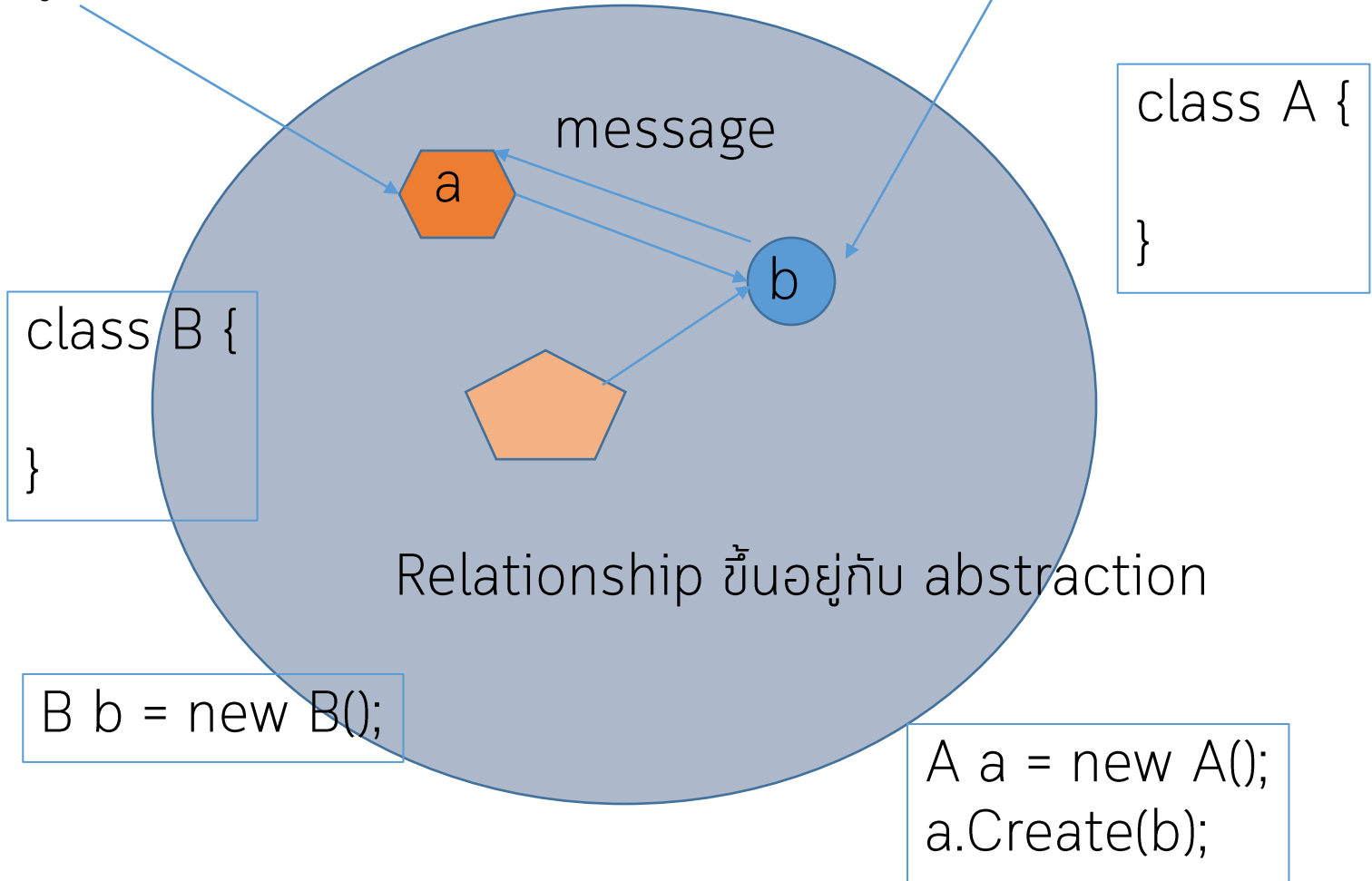
```
B b = new B();
```

```
A a = new A();  
a.Create(b);
```

# ระบบจองห้องพักโรงแรม

ลูกค้า

พนักงาน



# ระบบจองห้องพักโรงแรม

**ลูกค้า** ถามพนักงาน ว่ามี**ห้อง**ว่างหรือไม่

ลูกค้า ต้องการจองห้องพัก

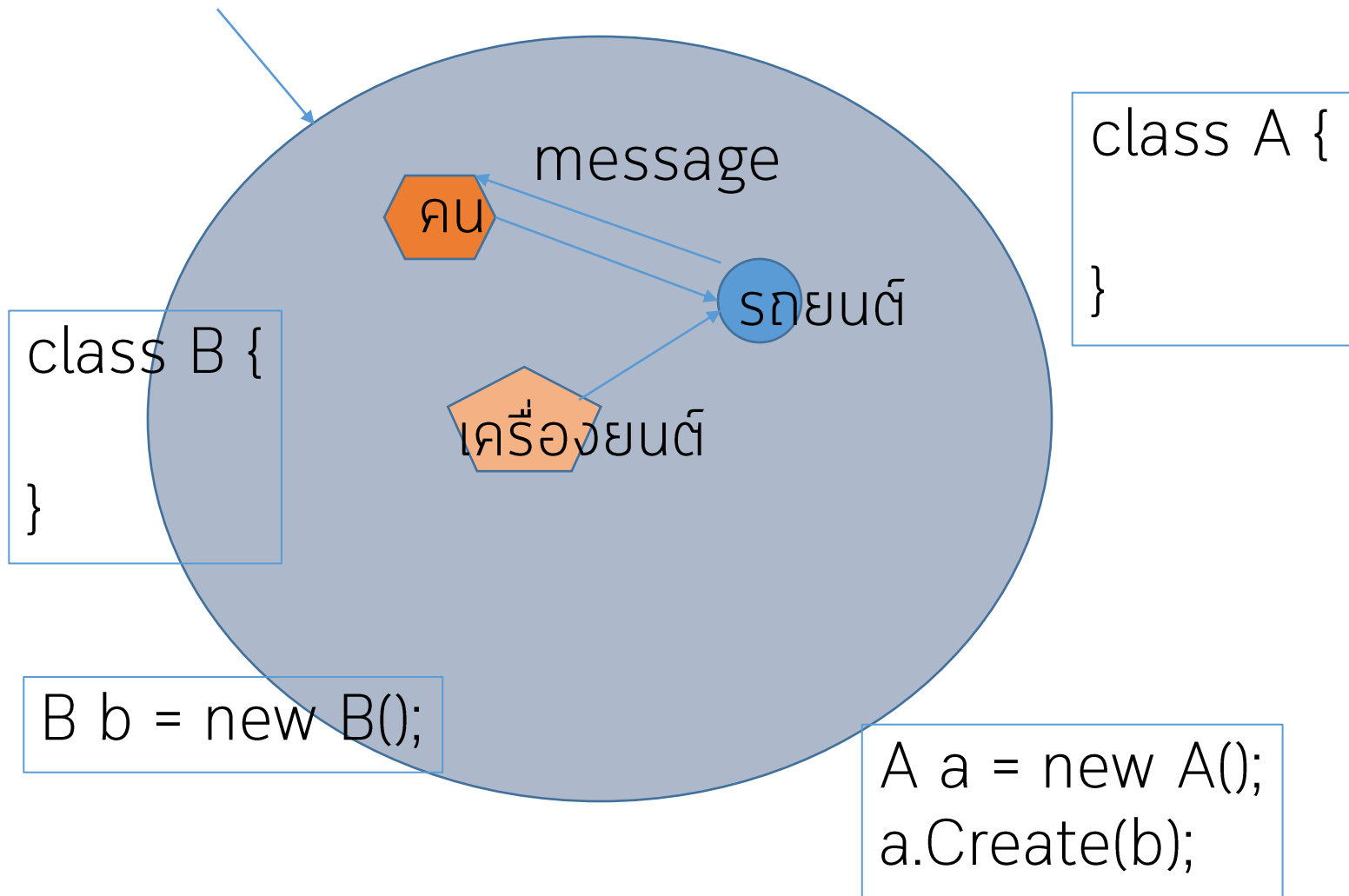
ลูกค้าต้องการเปลี่ยนชื่อการจอง ทำโดยการแจ้ง **พนักงาน**

พนักงานจะได้รับ message

จากลูกค้า

Interaction ปฏิสัมพันธ์

# Problem domain ขอบเขตสนใจ



Relationship ขึ้นอยู่กับ abstraction

เครื่องเป็นส่วนหนึ่งของรถยนต์ ใช้ aggregation abstract

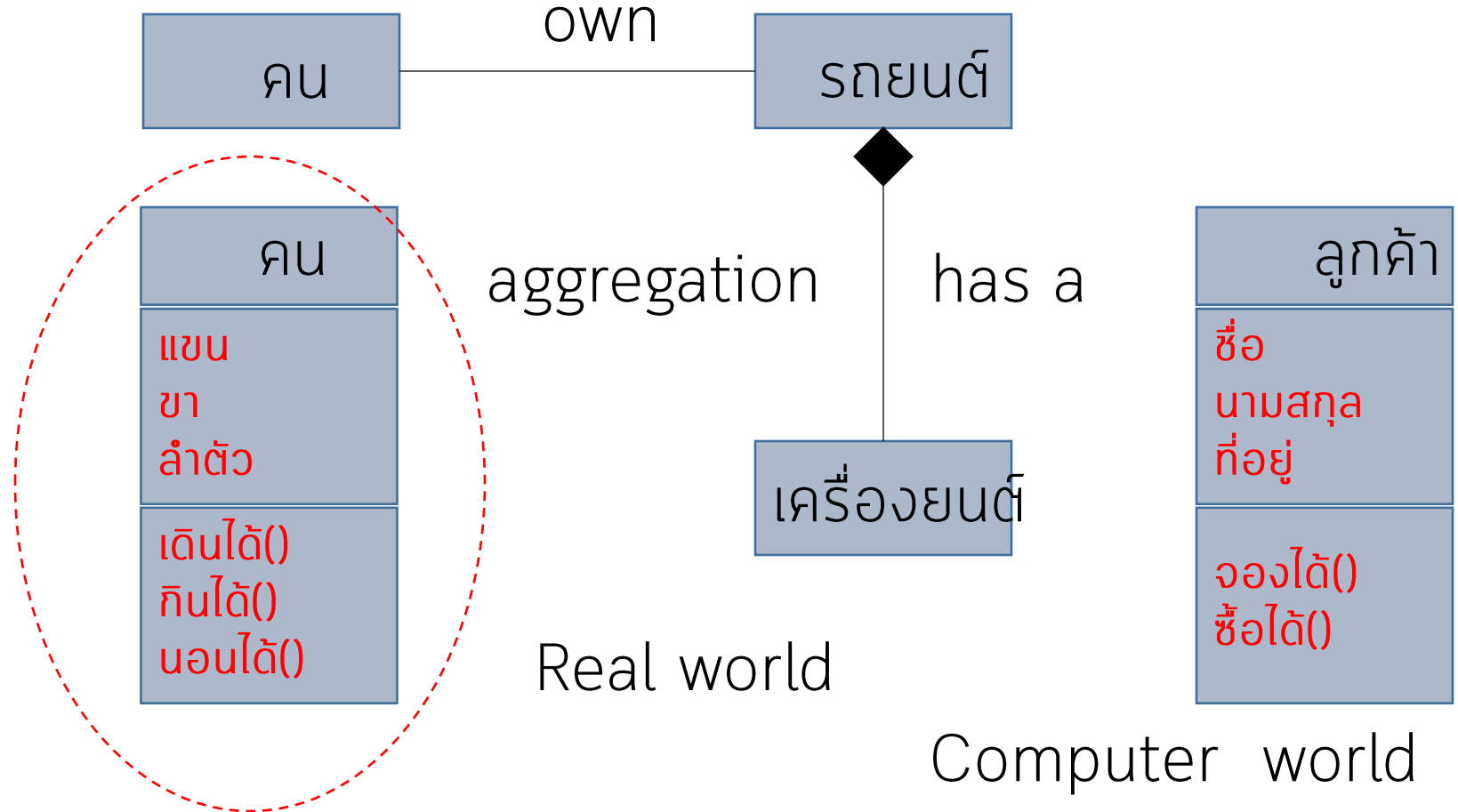


# Relationship

- คนขับรถยนต์ เป็น .....interaction
- คนเป็นเจ้าของรถยนต์ เป็น Relationship
- คน สตาร์ทรถยนต์ เป็น interaction
- คนนั่งในรถยนต์ Relationship.... Interaction

# Abstraction

association

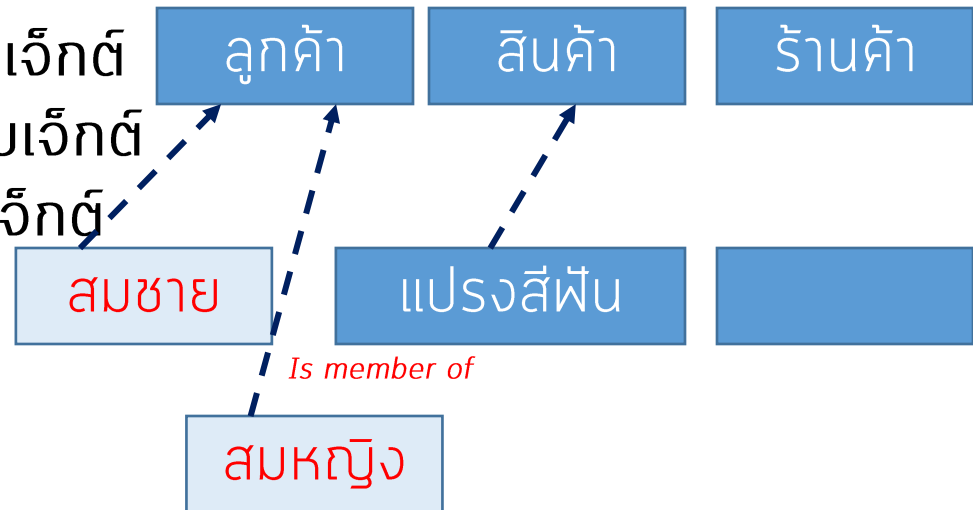


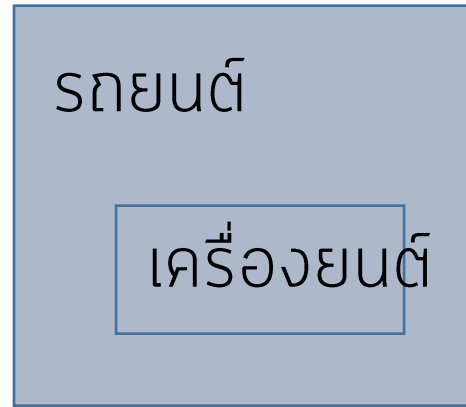
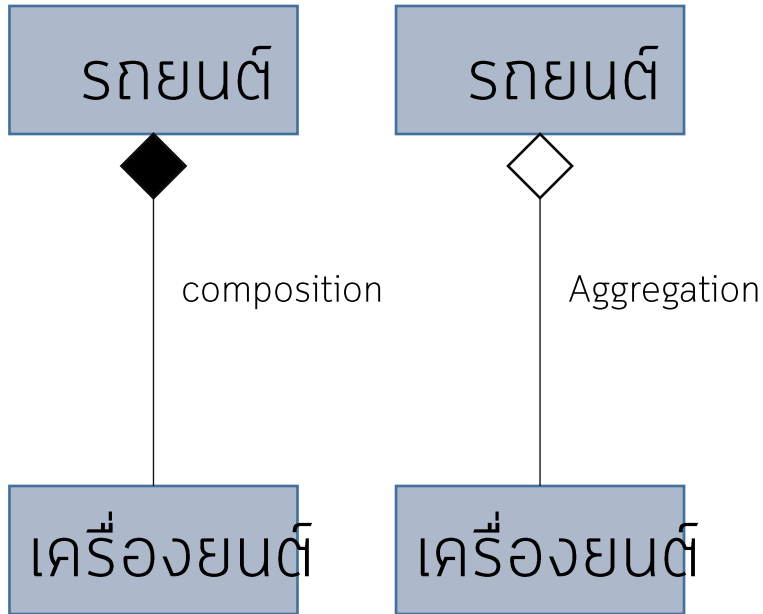
# สมาชิกไปซื้อแปรงสีฟัน ที่เซเว่น

## • มีกี่คลาส ที่ออบเจกต์

- ก. 1 คลาส 1 ออบเจกต์
- ข. 2 คลาส 2 ออบเจกต์
- ค. 3 คลาส 3 ออบเจกต์
- ง. 3 คลาส 2 ออบเจกต์
- จ. 2 คลาส 3 ออบเจกต์
- ฉ. ไม่ข้อใดถูกต้อง

Classification abstraction

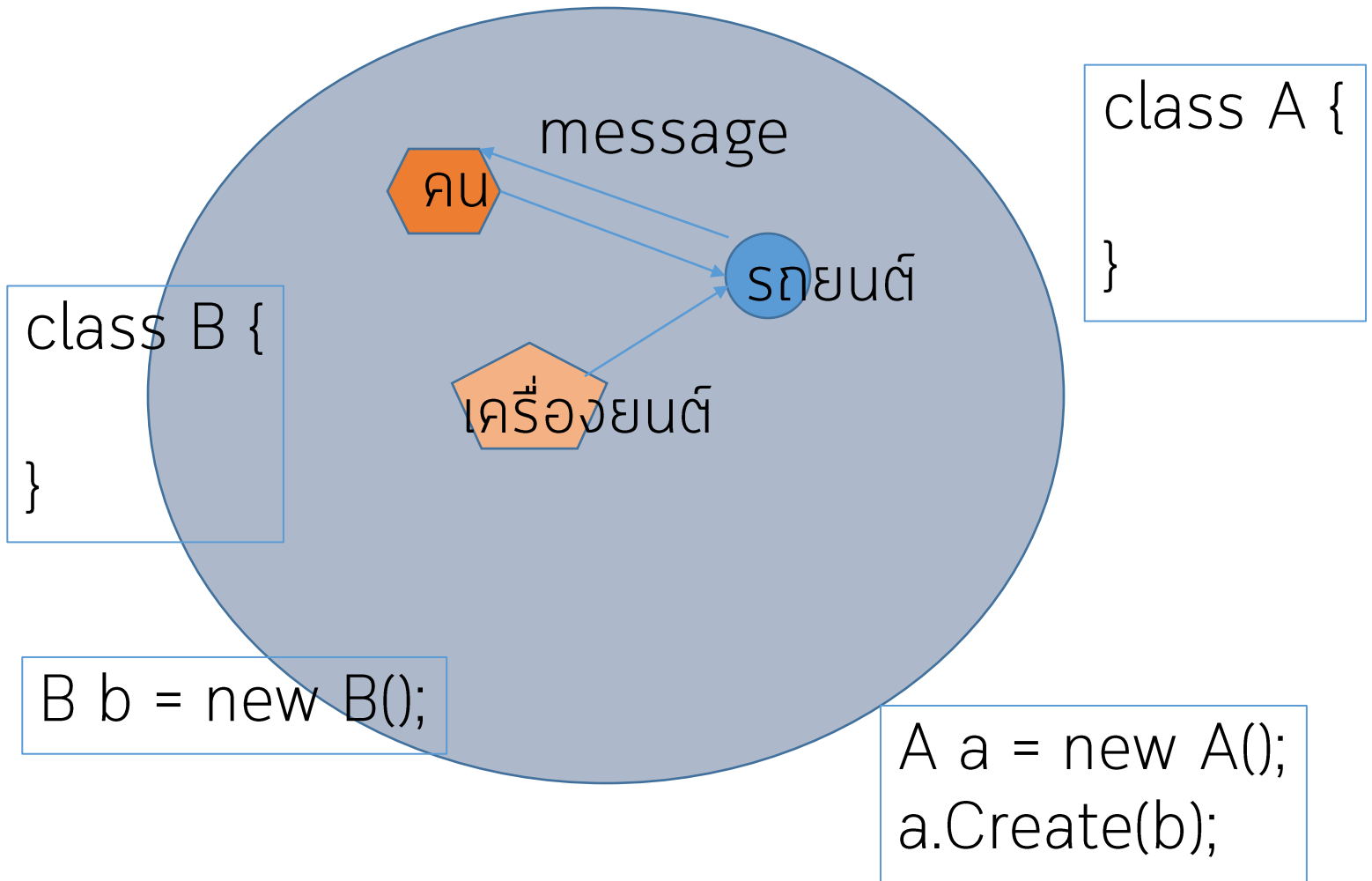




Inner class

```
class car{
  class engine{
  }
}
```

```
class car{
  Engine e = new Engine()
}
class Engine{
}
```



Relationship ขึ้นอยู่กับ abstraction  
 คนเป็นเจ้าของรถยนต์ ใช้ association abstraction

# สิ่งที่ฉลาดที่สุดในโลกนี้

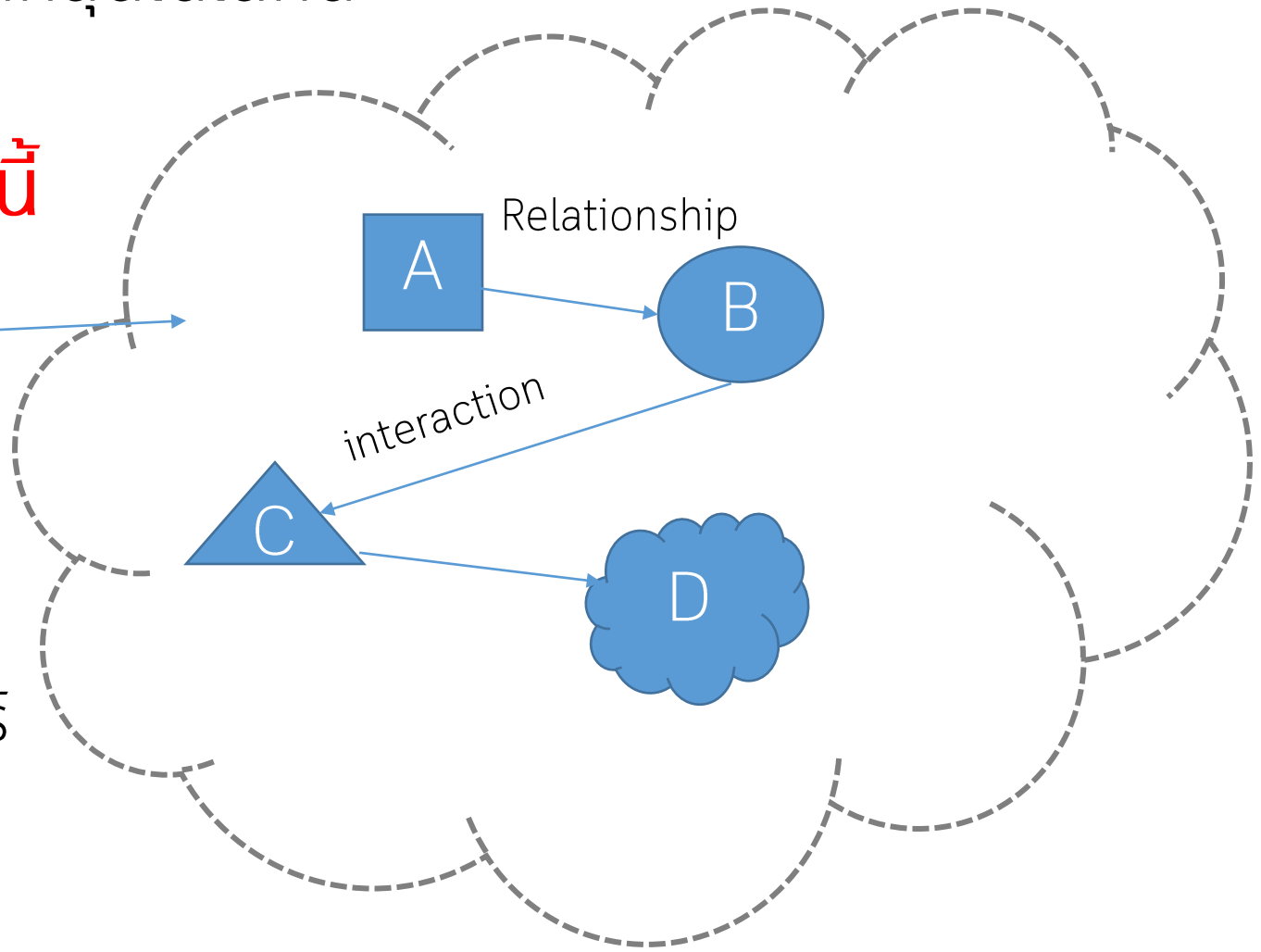
- ในโลกนี้

คน

คน

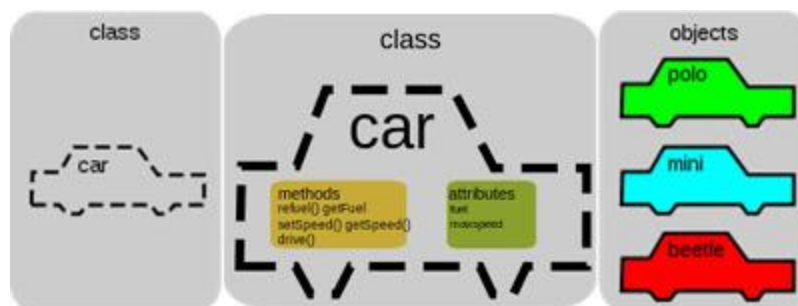
คอมพิวเตอร์

ธรรมชาติ



# Object Orientation

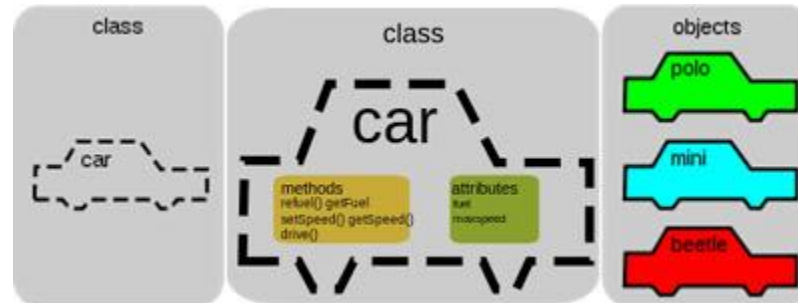
การมองทุกสิ่งในโลกให้เป็นวัตถุ เป็นการทำให้ปัญหาที่เป็นนามธรรม (Abstraction) กลายมาเป็นรูปธรรม เพื่อให้คอมพิวเตอร์สามารถแก้ปัญหาเหล่านั้นได้ด้วยการพัฒนาเป็นซอฟต์แวร์ และการพัฒนาซอฟต์แวร์ที่มีการนำเทคโนโลยี Object มาใช้งานก็คือ การโปรแกรมเชิงวัตถุ (Object - Oriented Programming) นั่นเอง



# Object - Oriented Programming (OOP)

## Object - Oriented Programming (OOP)

หรือ การโปรแกรมเชิงวัตถุ หมายถึง การเขียนโปรแกรมที่ประกอบไปด้วยกลุ่มของ Object ซึ่งแต่ละ Object จะถูกจัดกลุ่มอยู่ในรูปของคลาส (Class) โดยที่แต่ละคลาสสามารถมีการถ่ายทอดลักษณะ (Inheritance) กันลงมาอย่างเป็นลำดับ (Hierarchy)





# คุณสมบัติการเขียนโปรแกรมเชิงวัตถุ

1. ความสามารถในการสืบทอด (Inheritance)
2. ความสามารถในการห่อหุ้ม(Encapsulation)
3. ความสามารถในการแปลงร่าง/การพ้องรูป (Polymorphism)
4. ความสามารถในการจัดการโครงสร้างข้อมูลแบบเชิงนามธรรม (Abstract)

ถ้าเราจะเรียกภาษาใดว่าเป็นภาษาเชิงวัตถุ จะต้องประกอบไปด้วยคุณสมบัติ 4 ข้อข้างบน

Java , C++ , C# , VB.NET , Delphi , PHP , Python

# ความหมายของวัตถุ (Object)

**วัตถุ (Object)** คือ ตัวแทนของบุคคล สถานที่ หรือ สิ่งของที่น่าสนใจในเหตุการณ์หนึ่ง ซึ่งอาจจับต้องได้หรือจับต้องไม่ได้



# ประเภทของวัตถุ

- Tangible Object : มองเห็นได้
  - คน สิ่งของ สัตว์ ฯลฯ
- Intangible Object : มองไม่เห็น
  - บทบาท กฎเกณฑ์ กฎหมาย ภาพยนตร์ เนื้อหา เนื้อเรื่อง ซอฟต์แวร์

# ความหมายของวัตถุ

วัตถุ(Object) เป็นองค์ประกอบสำคัญของการเขียนโปรแกรมที่สามารถรับค่าและแสดงส่งค่าได้ และจากนิยาม Object คือตัวแทนของสิ่งของที่อยู่ในโลกของความเป็นจริง ซึ่งอาจเป็นสิ่งที่จับต้องได้หรือไม่ได้ จะต้องมีความลักษณะ (Attribute) ที่บ่งบอกว่าเป็น Object ของอะไร และมีพฤติกรรม (Behavior) ที่บ่งบอกให้ทราบว่า Object นั้นทำอะไร ดังนั้น Object จะสามารถตอบสนองต่อสิ่งที่กระทำในการร้องขอสิ่งต่างๆ ได้

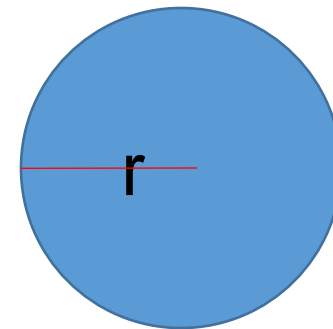
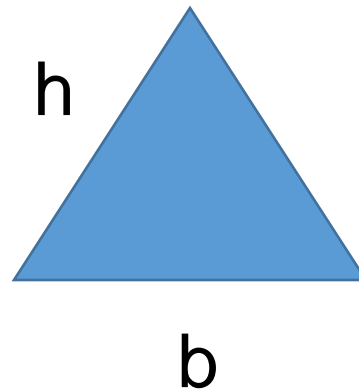
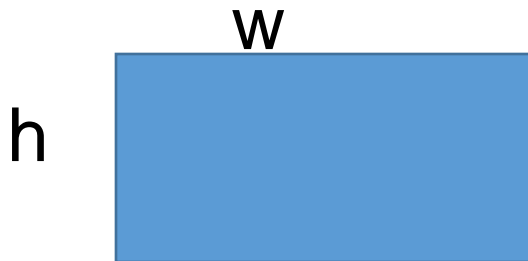
วัตถุ

- Attributes
- Function

## Tangible

เก้าอี้  
อาจารย์  
นักศึกษา  
คอมพิวเตอร์  
แอปเปิล

Attribute ของวัตถุมีได้ 2 อย่าง  
คือที่เป็นวัตถุ(ADT)  
และเป็นข้อมูลพื้นฐาน (PDT) เช่น  
เก้าอี้ ขาเก้าอี้ ขาเป็น object  
ราคา เป็น PDT เป็นข้อมูลพื้นฐาน



car

Primitive Data Type

engine : integer , String



car

engine: **Engine**

Abstract Data Type , Class

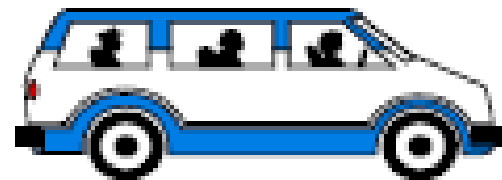
Engine

Model : String  
Size : integer  
Id : String

Start()  
Stop()

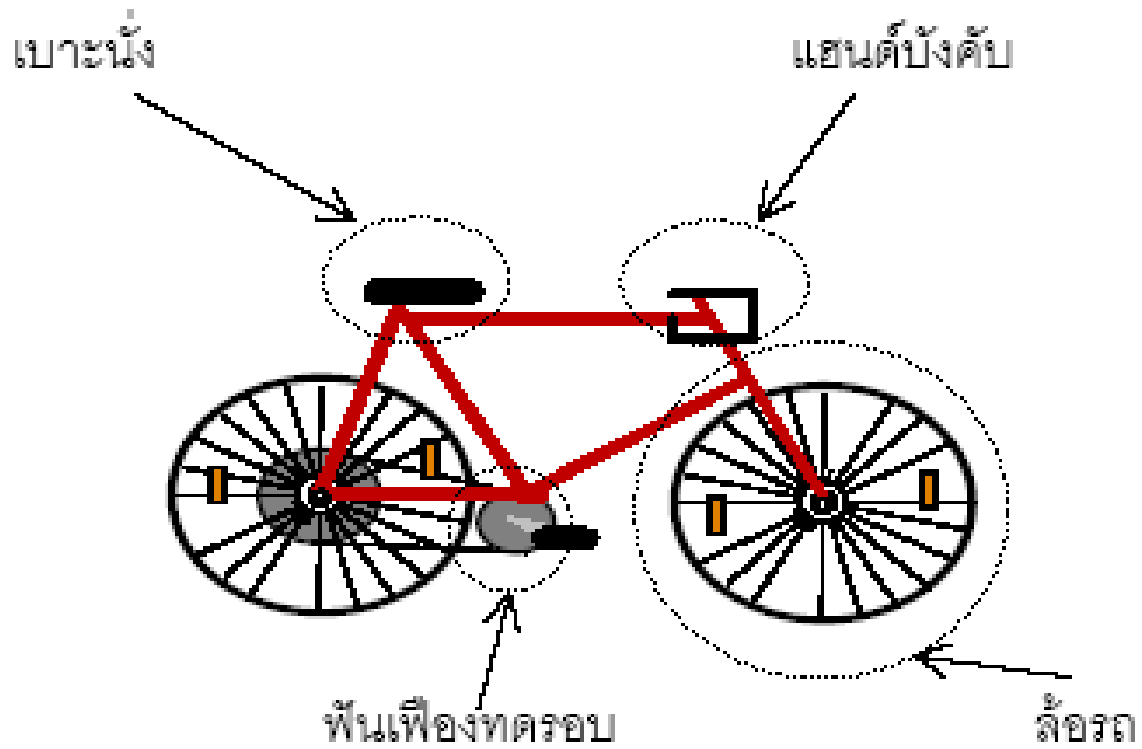


คอมพิวเตอร์



รถยนต์

รูปที่ 1 แสดงวัตถุต่างๆ ในโลกความเป็นจริง



รูปที่ 2 แสดงวัตถุรถจักรยานและวัตถุประกอบของรถจักรยาน



# ลักษณะที่วัตถุจะต้องมี

1. **Identity** คือ เอกลักษณ์เฉพาะตัวที่ทำให้ Object ของคลาสหนึ่งๆ แตกต่างจาก Object ของอีกหนึ่งคลาสหนึ่ง ซึ่งในภาษาจาวา identity จะหมายถึง Attribute หรือตัวแปร (Variable)

## ลักษณะที่วัตถุจะต้องมี

**2. State** คือสถานะของ Object ณ เวลาหนึ่งๆ ซึ่งสามารถเปลี่ยนแปลงได้ เมื่อมีการกระตุ้นจาก Object อื่นโดยการส่ง Message ในภาษาจาวา State จะหมายถึงค่าของ Attribute (Value of Attribute)

## ลักษณะที่วัตถุจะต้องมี

3. **Behavior** คือ พฤติกรรมที่ Object กระทำและตอบสนองทุกครั้ง โดยการกระตุ้น Object อื่นโดยการส่ง Message ต่อกันในภาษาจาวา Behavior ก็คือ *Method*

# โทรศัพท์มือถือ

- Identity / Value of Attribute ... สีแดง iPhoneX , ขนาด 7”
- State / Attribute ... Color , Price , Size .....
- Behavior / Function ... Call , Receive , SMS , Line

# ตู้เย็น

- Identity / Value of Attribute ... ..
- State / Attribute ... ..
- Behavior / Function ... ..

# Object & Class

- Class คือ กลุ่ม ของวัตถุ หรือ แม่แบบของวัตถุ หรือ การ จัดวัตถุชนิด ให้อยู่ในกลุ่มเดียวกัน
- Object คือ การระบุ เจาะจง ลงไป

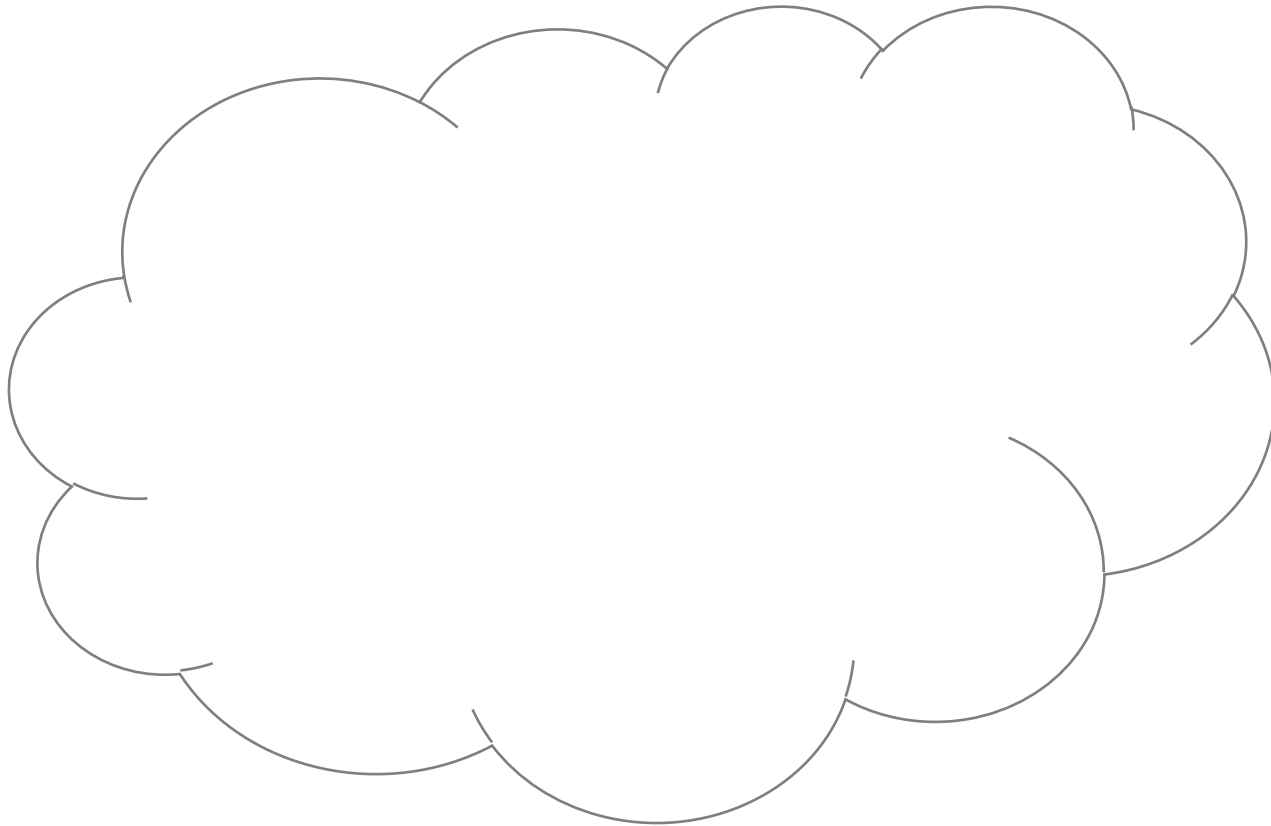
- โทรศัพท์มือถือ เป็นคลาส
- Samsung s7 , iPhoneX เป็น object ของคลาส โทรศัพท์มือถือ

ในโลกความเป็นจริง วัตถุ มีเยอะมากมาย ไม่สามารถเรียกได้หมด จึงตั้งชื่อกลุ่ม เพื่อใช้เรียกสิ่งๆที่เหมือนกัน เรียกว่า คลาสเดียวกัน

ต้นไม้

สัตว์

คน



# แนวคิดในการเขียนโปรแกรม (Programming Concepts)

ในทาง OOP คุณสมบัติของ Object (identity + behavior) ที่คล้ายกันจะถูกจัดไว้ในคลาสเดียวกัน ดังนั้นการแยกแยะ Object ที่สร้างจากคลาสเดียวกันจึงใช้ State ในการพิจารณา เช่น ชื่อของ Object (ค่าของตัวแปร “name”) เป็นต้น ส่วนการแยกแยะ Object ที่สร้างจากคลาสต่างกัน จะใช้ identity และ Behavior ในการพิจารณาเพราะ identity และ behavior ของ Object ต่างกันทำให้จัดอยู่ต่างคลาสนั้นนั่นเอง

# แนวคิดในการเขียนโปรแกรม (Programming Concepts)

แนวความคิดที่ใช้ในการเขียนโปรแกรมในปัจจุบันจะใช้ 2  
แนวความคิดหลักๆ ได้แก่ แนวความคิดการเขียนโปรแกรมแบบ  
โครงสร้าง และแนวความคิดเชิงวัตถุ



## แนวความคิดการเขียนโปรแกรมแบบโครงสร้าง (Structured Programming Concept)

แนวความคิดนี้เป็นการจัดการคำสั่งต่าง ๆ ใหม่อะไรที่มีรูปแบบและมาตรฐานที่สามารถเขียนโปรแกรมได้ง่าย อีกทั้งยังมีการตรวจสอบการทำงานของโปรแกรมได้โดยไม่ยุ่งยาก เมื่อเขียนโปรแกรมก็เหมือนกับการเขียนหนังสือหรือการอ่านหนังสือ คือมีจุดเริ่มต้นจากซ้ายไปขวา และจากบนลงล่างเหมือนการอ่านหนังสือ

# แนวความคิดเชิงวัตถุ (Object-Oriented Programming Concept)

เป็นการเขียนโปรแกรมเพื่อให้เป็นไปตามหลักการเชิงวัตถุ โดยผู้เขียนโปรแกรมต้องพยายามมองรูปแบบวัตถุให้ออกกอน ซึ่งต้องอาศัยจินตนาการพอสมควร

# แนวความคิดเชิงวัตถุ (Object-Oriented Programming Concept)

- แนวคิดการเขียนโปรแกรมแบบดั้งเดิมมักนิยมใช้ การเขียนโปรแกรมเชิงกระบวนการ (Procedural Programming) ซึ่งให้ความสำคัญกับขั้นตอนกระบวนการที่ทำ โดยแบ่งโปรแกรมออกเป็นส่วนๆตามลำดับขั้นตอนการทำงาน แต่แนวคิดการเขียนโปรแกรมเชิงวัตถุให้ความสำคัญกับ ข้อมูล (data) และ พฤติกรรม (behavior) ของวัตถุ และความสัมพันธ์กันระหว่างวัตถุกันมากกว่า

# เปรียบเทียบแนวคิดระหว่างการเขียน โปรแกรมเชิงกระบวนการ และเชิงวัตถุ

# ตัวอย่างตู้ขายเครื่องดื่มอัตโนมัติ



# วิธีการคิดแบบการเขียนโปรแกรมเชิงกระบวนการ

เมื่อมีการหยอดเหรียญเข้าตู้:

- ตรวจสอบจำนวนเหรียญและชนิดของเหรียญ
- แสดงผลชนิดของน้ำที่สามารถเลือกซื้อได้
- ตรวจสอบจำนวนน้ำกระป๋องที่มีอยู่ในตู้
- รับผลการเลือกชนิดน้ำ
- ส่งน้ำที่เลือกออกมาจากช่อง
- จัดเก็บเงินเข้าระบบ
- หากมีเงินทอน ให้ทอนเงินที่เหลือ ที่ช่องรับเงินทอน

# วิธีการคิดแบบการเขียนโปรแกรมเชิงวัตถุ

- ตู้ขายเครื่องดื่มอัตโนมัติ ประกอบด้วยส่วนประกอบต่างๆ ได้แก่ หน่วยตรวจสอบและจัดการเรื่องเงิน หน่วยจัดการเครื่องดื่ม หน่วยแสดงผลและรอรับคำสั่ง
- หน่วยตรวจสอบและจัดการเรื่องเงิน มีข้อมูลเกี่ยวกับเงินที่ได้รับ และเงินที่มีอยู่ในระบบ สามารถรับและตรวจสอบเงินที่หยอดเข้ามาได้ และทอนเงินได้
- หน่วยจัดการเครื่องดื่ม มีข้อมูลชนิดของเครื่องดื่ม จำนวนเครื่องดื่ม สามารถจัดเตรียมชนิดเครื่องดื่มที่พอกับเงินที่หยอด และสามารถจ่ายเครื่องดื่มออกมาจากตู้ได้
- หน่วยแสดงผลและรอรับคำสั่ง มีหน้าที่รอรับคำสั่ง และแสดงผลเงินที่หยอดเข้ามา

# หลักการทำงาน

- หน่วยจัดการเครื่องดืม มีข้อมูลชนิดของเครื่องดืม จำนวนเครื่องดืม สามารถจัดเตรียมชนิดเครื่องดืมที่พอกับเงินที่หยอด และสามารถจ่ายเครื่องดืมออกมาจากตู้ได้

|                       |
|-----------------------|
| หน่วยจัดการเครื่องดืม |
| ชนิดเครื่องดืม        |
| เงินทอน               |
| จำนวนเครื่องดืม       |
| ทอนเงิน()             |
| จ่ายเครื่องดืม()      |



# จงเขียนขั้นตอนการทำงานของตู้ทอนเงิน

- ต.ย. สมมุติ ราคา สินค้า 125.75 จ่ายแบงค์หนึ่งพันบาท (1000) จะต้องทอนออกมาเป็น 874.25 บาท

- แบงค์ 500 จำนวน 1 ฉบับ
- แบงค์ 100 จำนวน 3 ฉบับ
- แบงค์ 50 จำนวน 1 ฉบับ
- แบงค์ 20 จำนวน 1 ฉบับ
- เหรียญ 10 จำนวน 0 เหรียญ
- เหรียญ 5 จำนวน 0 เหรียญ
- เหรียญ 1 จำนวน 4 เหรียญ
- เหรียญ .50 จำนวน 0 เหรียญ
- เหรียญ .25 จำนวน 1 เหรียญ

# ขั้นตอนวิธี

## กลุ่ม 1

1. ตรวจสอบจำนวนเงินและชนิดของเงินที่รับมา
2. ตรวจสอบราคา และ จำนวนสินค้า
3. คำนวณจำนวนเงินและแสดงผลจำนวนเงินทั้งหมด
4. แสดงจำนวนเงินก่อนทั้งหมด
5. ทอนเงินโดยเลือกจากชนิดและจำนวนเงินมากที่สุดก่อน
6. ทอนเงินและชนิดเงินตามเศษที่คำนวณแล้ว
7. จ่ายสินค้าตามรายการที่เลือก

# *GIGO: Gabag In Gabag out*

## Algorithm กฎข้อแรก

1. ต้องมีเริ่มต้น
2. ....
3. ....
4. ต้องมีสิ้นสุด

**START**

**STOP**

Algorithm ง่ายๆ สอง 1 บรรทัด 1 งาน 1 job สั้น  
กะทัดรัด ได้ใจความ หรือ ที่คอมฯสามารถทำงานได้

START

1. ต้องมีเริ่มต้น
2. ....
3. ....
4. ต้องมีสิ้นสุด

STOP

# Algorithm

ขั้นตอนแรก ต้อง

รับจำนวนเงินเข้ามา

รับ ราคาสินค้า ที่ต้องจ่าย

คิด เงินที่ต้อง ทอน คิดยังไง .....

- INPUT
- PROCESS
- OUTPUT

เงินทอน = เงินที่รับเข้ามา - ราคาสินค้า

Exchange = Money - Price

# Algorithm

- 874.75 ทำอย่างไรถึงจะหาแบงค์ 500

1

$$\begin{aligned} B500 &= \text{Exchange} / 500 \\ \text{Tmp} &= \text{Exchange} \% 500 \end{aligned}$$

274.75

เศษที่เหลือ

MOD  
%

/ DIV ไม่สนใจเศษ

\ MOD เอาเศษที่เหลือ

# Algorithm

- 274.75 ทำอย่างไรถึงจะหาแบงค์ 100

1

$$\begin{aligned} B100 &= \text{Exchange} / 100 \\ \text{Tmp} &= \text{Exchange} \% 100 \end{aligned}$$

274.75

เศษที่เหลือ  
MOD  
%

/ DIV ไม่สนใจเศษ  
\ MOD เอาเศษที่เหลือ

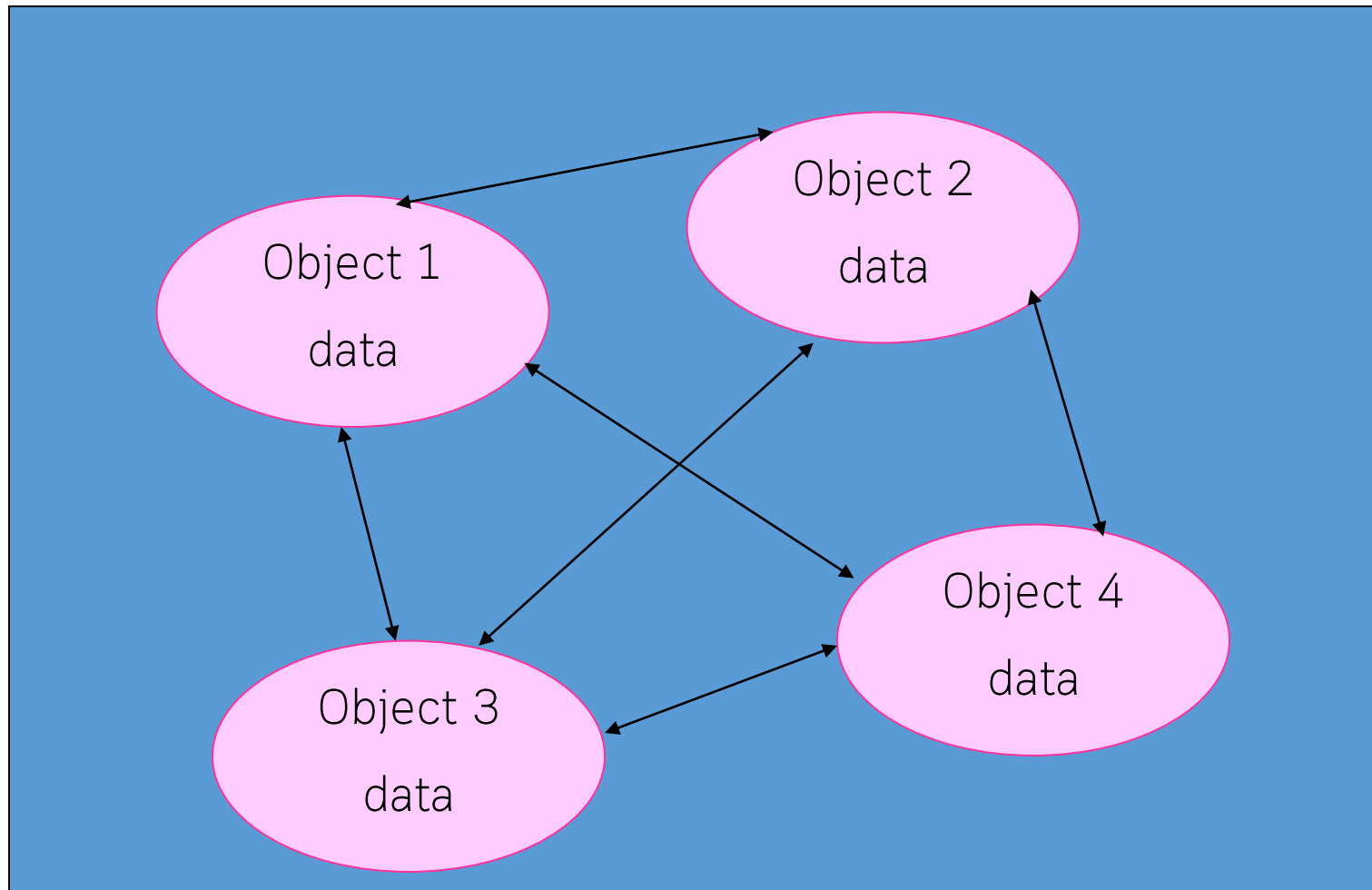
# การเขียนโปรแกรมเชิงวัตถุ (Object - Oriented Programming)

การเขียนโปรแกรมเชิงวัตถุ เป็นวิธีการเขียนโปรแกรมแบบใหม่ ซึ่งจะเรียกโปรแกรมย่อยที่ทำหน้าที่เฉพาะว่า Object และยังรวม data เข้าเป็นส่วนหนึ่งของ Object ด้วย นอกจากนี้โปรแกรมเชิงวัตถุยังมีโครงสร้างพิเศษนอกเหนือจาก Procedure Programming หรือ Module Programming คือ



# การเขียนโปรแกรมเชิงวัตถุ (Object - Oriented Programming)

การทำงานของแต่ละ Object จะไม่สิ้นสุดลง เมื่อกระบวนการทำงานของโปรแกรมจบการทำงานบน Object นั้นแล้ว ซึ่ง Object ที่อยู่ในโปรแกรมเหล่านี้ พร้อมเสมอในการเริ่มทำงานต่อตำแหน่งเดิมเมื่อเกิดการเรียกใช้งาน Object อีกครั้ง ซึ่งแตกต่างจากการเขียนโปรแกรมในแบบอื่นๆ ที่โปรแกรมย่อยจะต้องเริ่มการทำงานใหม่เมื่อถูกเรียกใช้อีกครั้ง ดังนั้นเทคนิคนี้จึงช่วยลดปัญหาที่เกิดขึ้นจากการเขียนโปรแกรมด้วยวิธีอื่นๆ ได้



แต่ละ Object สามารถติดต่อสื่อสารกันได้โดยการส่ง Message ไปที่ Object อื่น

แนวคิดที่สำคัญของ  
การเขียนโปรแกรมเชิงวัตถุ

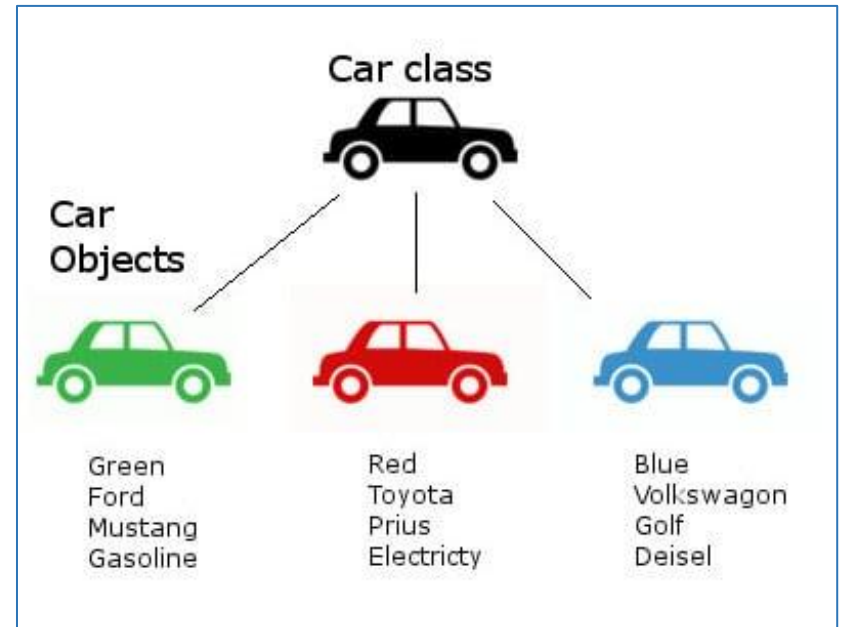
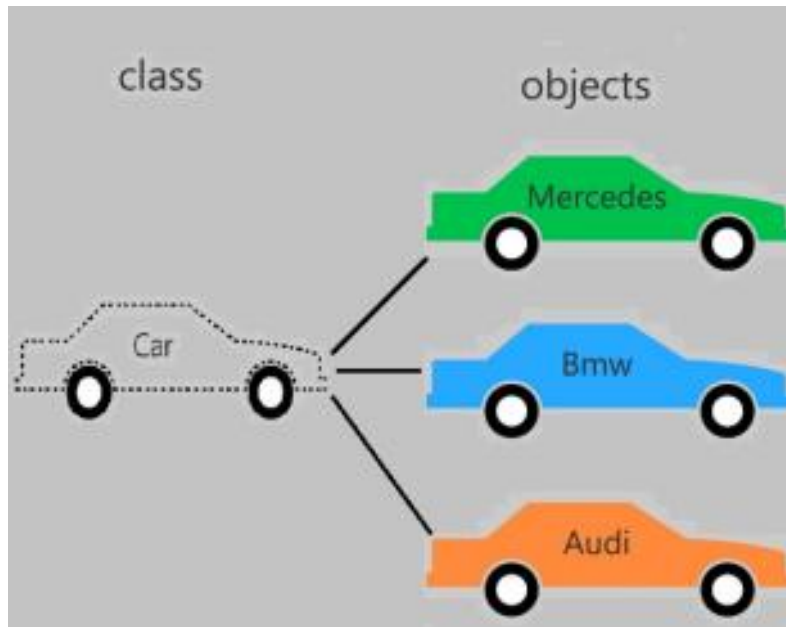
# แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

## Class

- คลาส (Class) - ประเภทของวัตถุ เป็นการกำหนดว่า วัตถุจะประกอบไปด้วย ข้อมูล(data) หรือคุณสมบัติ(property) และ พฤติกรรม(behavior)หรือการกระทำ(method) อะไรบ้าง ซึ่ง คลาส (เช่น มนุษย์) เป็นโครงสร้างพื้นฐานของการเขียนโปรแกรมเชิงวัตถุ

# แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

## Class and Objects



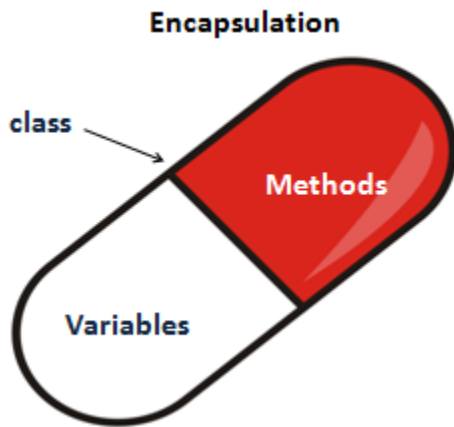
# แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

## Object

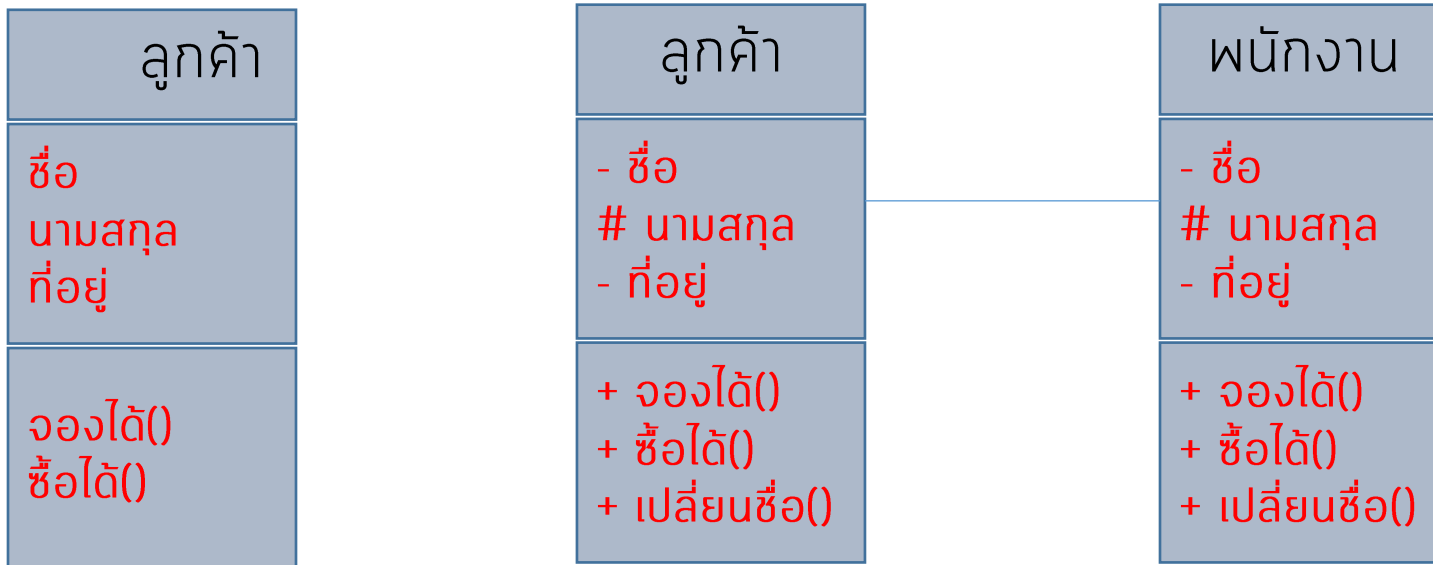
- วัตถุ (Object) - โดยมากจะเรียกว่า ออบเจกต์ คือ ตัวตน (instance) ของ คลาส(เช่น นายทักษิณ, นายสนธิ) ซึ่งจะเกิดขึ้นระหว่าง run-time โดยแต่ละ ออบเจกต์ จะมีข้อมูลเฉพาะของตัวเอง ทำให้ ออบเจกต์ แต่ละ ออบเจกต์ ของ คลาส ซึ่งใช้ source code เดียวกันมีคุณลักษณะและคุณสมบัติที่แตกต่างกัน

# แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

## Encapsulation



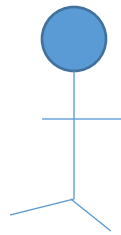
- **Encapsulation** - การปิดบังข้อมูล เป็นวิธีการกำหนดสิทธิ์ในการเข้าถึงข้อมูล หรือการกระทำกับออบเจกต์ ของ คลาสนั้นๆ ทำให้แน่ใจได้ว่าข้อมูลของออบเจกต์นั้นจะถูกเปลี่ยนแปลงแก้ไขผ่านทาง methods หรือ properties ที่อนุญาตเท่านั้น (เช่น การกำหนดตำแหน่งทางการเมือง เป็น public method ที่ผู้อื่นสามารถกระทำได้ ส่วนการลาออกจากตำแหน่ง เป็น private method ที่มีแต่ ออบเจกต์ ของ คลาส เท่านั้นที่จะสามารถทำได้ แต่การกดดันและการขบไล่สามารถสร้าง data ที่อาจจะส่งผลเกิดการลาออกได้เช่นกัน)



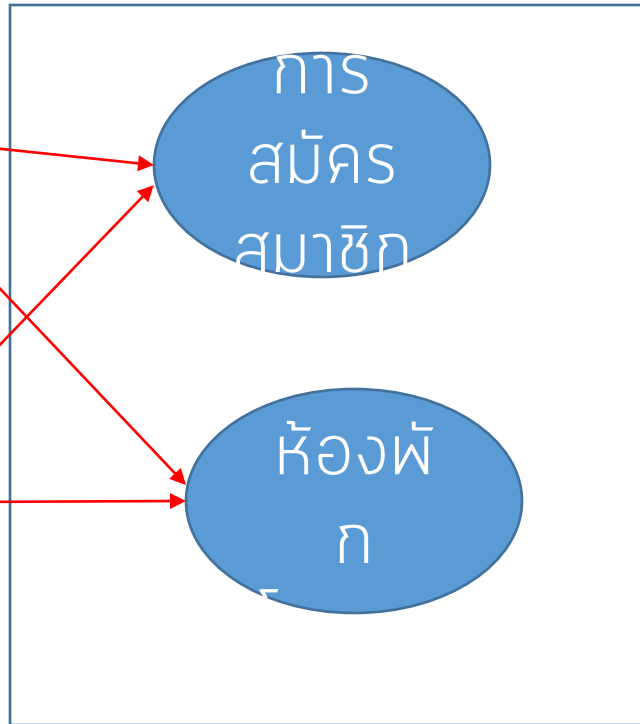
- Private  
 + Public  
 # Protected



ลูกค้า



พนักงาน



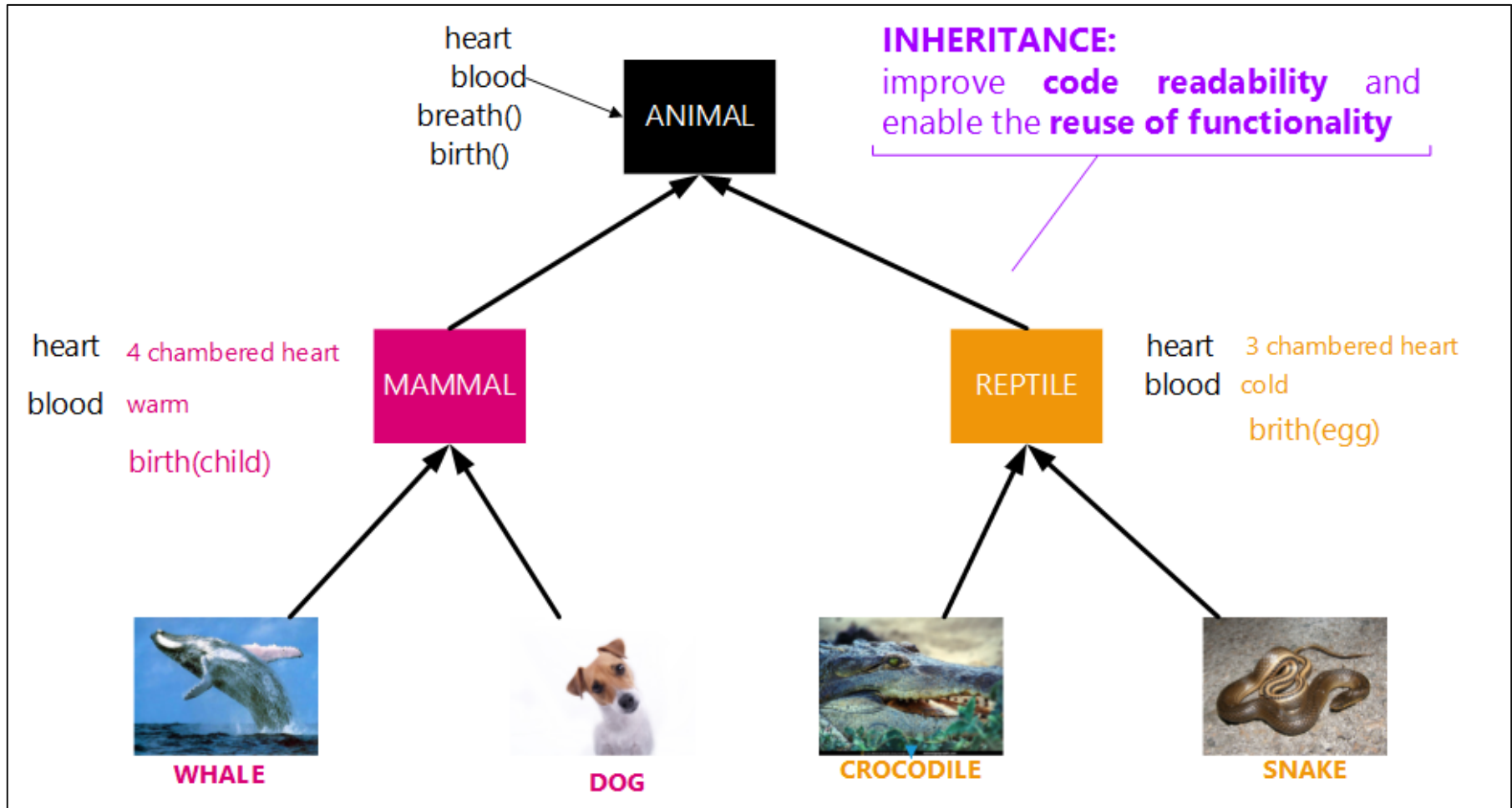
# แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

## Inheritance

- **Inheritance** - การสืบทอดคุณสมบัติ เป็นวิธีการสร้าง คลาสย่อย ที่เรียกว่าซับคลาส (subclass) ซึ่งจะเพื่อกำหนดประเภทของวัตถุให้จำเพาะเจาะจงขึ้น ซึ่ง ซับคลาส จะได้รับถ่ายทอดคุณสมบัติต่างๆมาจากคลาสหลักด้วย (เช่น คลาส มนุษย์ สืบทอดมาจาก คลาส สิ่งมีชีวิต)

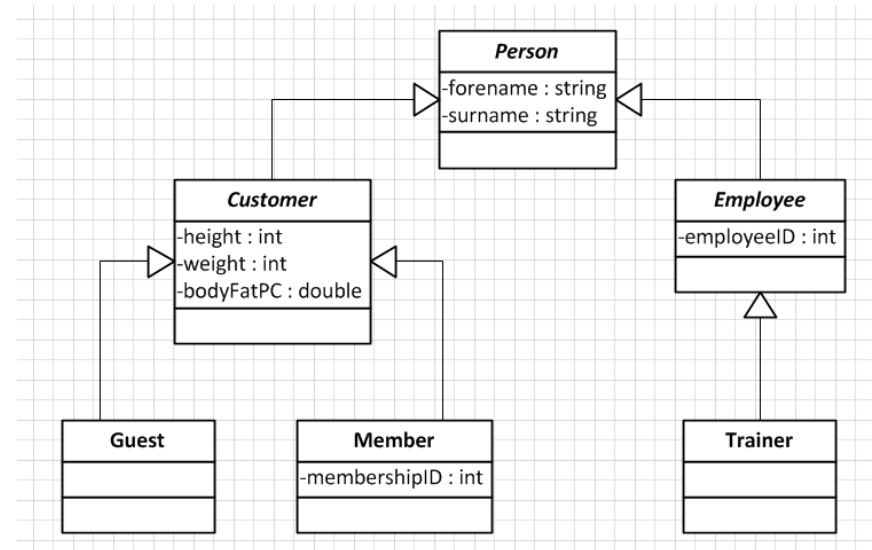
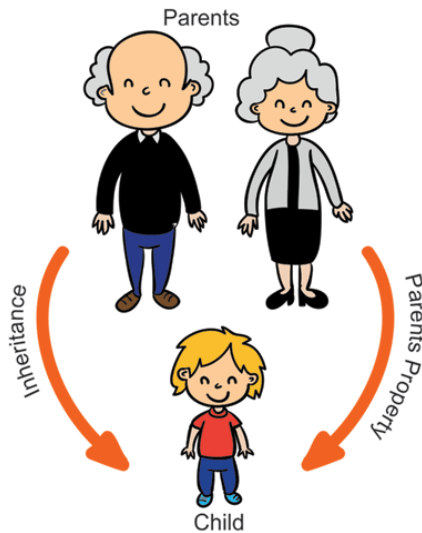
# แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

## Inheritance



# แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

## Inheritance



# แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

## Abstraction

- *Abstraction* - นามธรรม เป็นการแสดงถึงคุณลักษณะและพฤติกรรมของ object เท่าที่จำเป็นต้องรับรู้และใช้งาน โดยซ่อนส่วนที่เหลือเอาไว้เพื่อไม่ให้เกิดความสับสน เช่น ตามปกติแล้ว นาย**บัญชาพงศ์** จัดเป็นตัวแทนของ คลาส**มนุษย์** ซึ่งจะมีพฤติกรรม การกระทำทุกอย่างที่ตามที่กำหนดไว้ตามโครงสร้างของ คลาส มนุษย์ แต่ในบางกรณีที่น่าไปใช้งาน เราไม่ต้องการให้เกิดการสับสนต่อการใช้งานหรือการจัดประเภทมาก เราสามารถจัดการหรือใช้งาน อ็อบเจกต์ นายบัญชาพงศ์ ให้อยู่ในรูปของสิ่งมีชีวิต ก็ได้

# แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

## Polymorphism

- ***Polymorphism*** - ภาวะที่มีหลายรูปแบบ เป็นวิธีการกำหนดรูปแบบการกระทำที่เหมือนกันแต่ได้ผลที่แตกต่างกัน เช่น การแปลงเสียง เป็น method หลักของ คลาส สิ่งมีชีวิต ซึ่งมีคลาส มนุษย์ และคลาสสุนัข เป็น ชับคลาส แต่ผลของการแปลงเสียงของอ็อบเจกต์จากคลาสทั้งสอง จะออกมาไม่เหมือนกัน

# ตัวอย่างโปรแกรม

```
import javax.swing.*;  
import java.awt.*;  
public class CardView extends JLabel  
{  
    private ImageIcon icon;
```

```
public CardView(VCard card)
{
    getImage(card.getImage());
    setIcon(icon);
    setBackground(Color.white);
    setOpaque(true);
}
```



Object 1  
data



```
private void getImage(String name)
{
    java.net.URL url =
    this.getClass().getResource(name);
    icon = new ImageIcon(url);
}
}
```

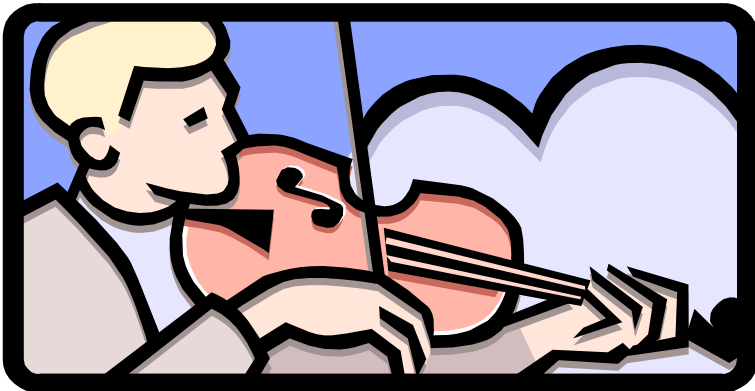


Object 2  
data

**...Break..**

# แนวคิดการเขียนโปรแกรมเชิงวัตถุ

เป็นการจำลองความเป็นจริงในชีวิตความเป็นอยู่ของมนุษย์ที่จะประกอบด้วย คน สัตว์ สิ่งของ และใช้สิ่งต่างๆ เพื่อแก้ปัญหา โดยแต่ละสิ่ง มีหน้าที่ในการแก้ปัญหานั้น ๆ



ต้องการหาความบันเทิงด้วยดนตรี ก็ต้องมีสิ่งต่างๆ คือเครื่องดนตรีและคนเล่นดนตรี เป็นต้น

# Structural Programming VS Object Oriented Programming

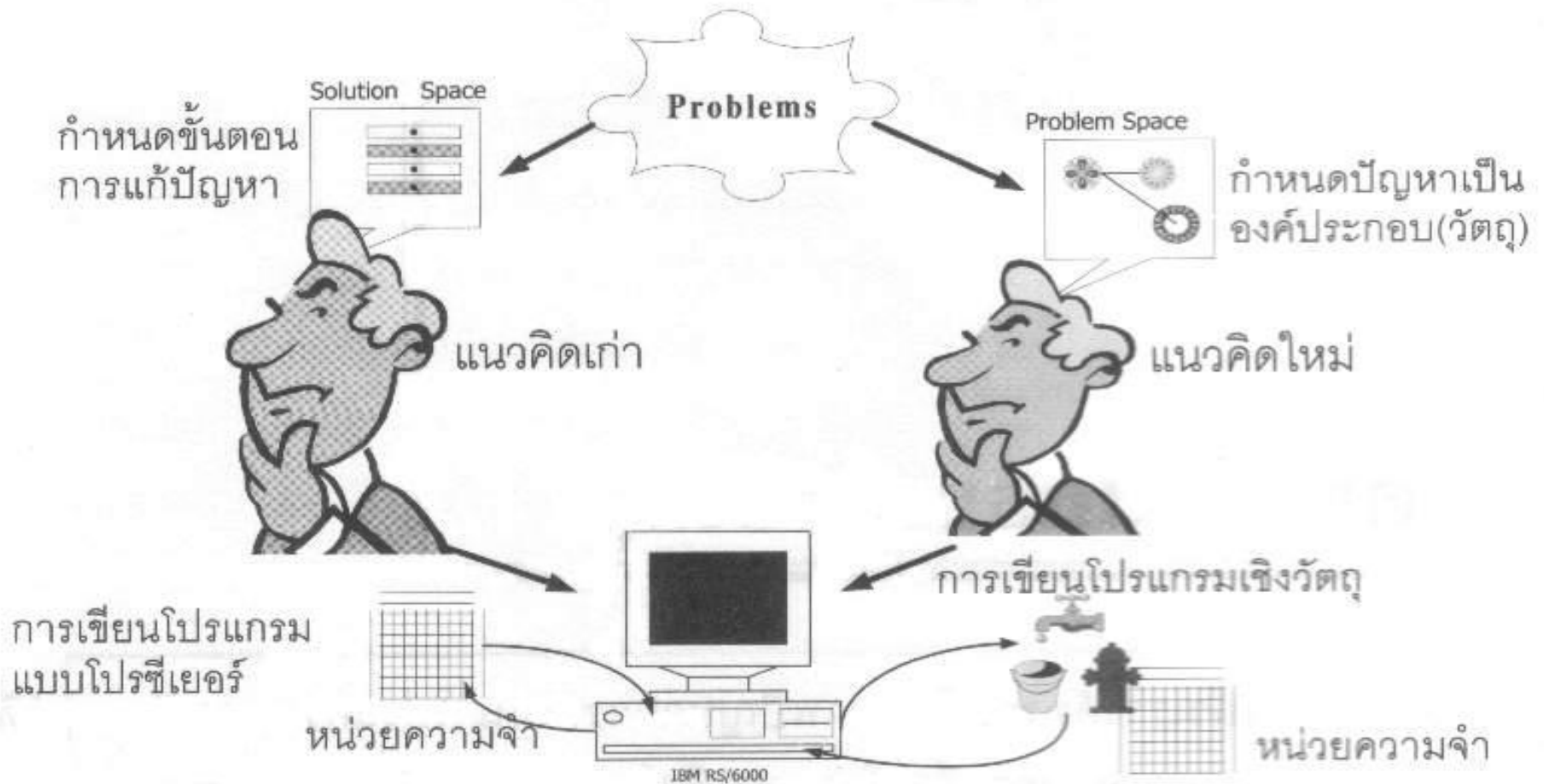
```
#include <stdio.h>
main () {
    int width , length, area ;
        width = 10 ;
        length = 5 ;
        area = width * length ;
        printf ("Area = %d\n",area) ;
return 0
}
```

- ▶ แก้ไขปัญหาด้วยโปรแกรมที่ทำตามลำดับขั้นตอนที่ได้กำหนดไว้
- ▶ มีการเรียกใช้โปรแกรมย่อยที่สร้างขึ้น



- ▶ แก้ปัญหาโดยมองสิ่งต่าง ๆ ที่อยู่ในโปรแกรมเป็นวัตถุ
- ▶ วัตถุแต่ละชิ้น มีคุณสมบัติ และความสามารถแตกต่างกันออกไป
- ▶ วัตถุติดต่อกันด้วยวิธีการส่งข่าวสารถึงกัน

# Structural Programming VS Object Oriented Programming



แนวคิดเก่าและใหม่ของการเขียนคำสั่งคอมพิวเตอร์

# แนวคิดของเอลัน เคย์ (Alan Kay)

Alan key เป็นผู้หนึ่งที่พัฒนาตัวแปร small Talk ซึ่งเป็นต้นแบบของการเขียนโปรแกรมเชิงวัตถุ ได้ให้นิยามของภาษาเชิงวัตถุไว้ดังนี้

- ทุกๆ สิ่งเป็นวัตถุ
- โปรแกรม คือ กลุ่มของวัตถุที่ส่งสารบอกกันเพื่อทำงาน
- วัตถุแต่ละวัตถุมีหน่วยความจำ
- วัตถุต้องมีชนิด
- วัตถุที่จัดอยู่ในประเภทเดียวกันย่อมได้รับข่าวสารเหมือนกัน

# ทุกๆ สิ่งเป็นวัตถุ

- องค์ประกอบของโปรแกรมคอมพิวเตอร์ทุก ๆ ส่วนจะต้องเป็นวัตถุซึ่งกฎข้อนี้เองที่ทำให้ล่ามแปลภาษาส่วนมากรวมทั้ง JAVA ด้วย ไม่เป็นภาษาเชิงวัตถุแบบบริสุทธิ์เพราะในล่ามแปลภาษา JAVA ยังมีบางส่วนที่ไม่เป็นวัตถุ เช่น การกำหนดตัวแปร เป็นต้น



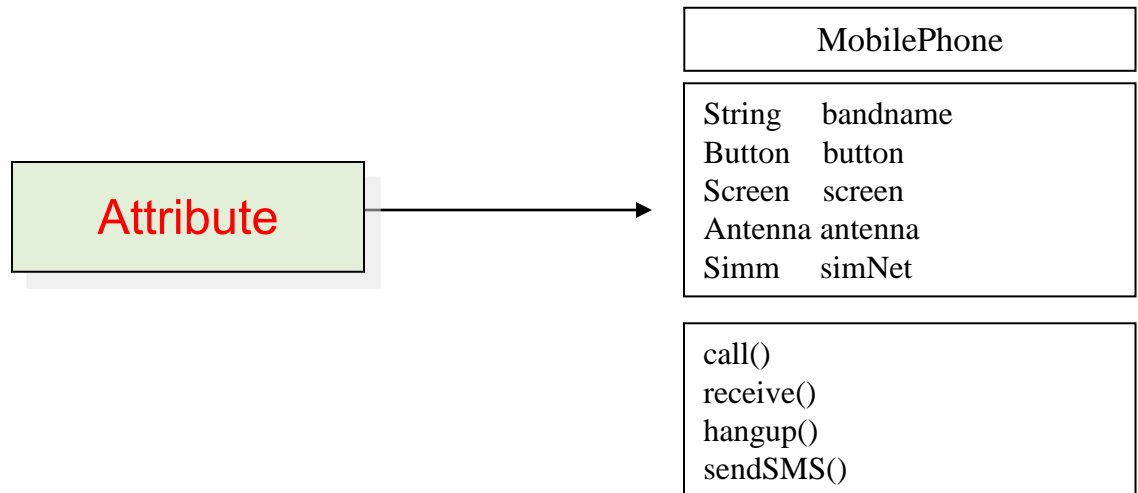
# โปรแกรมคือกลุ่มของวัตถุ

- โปรแกรมในความหมายของการเขียนโปรแกรมเชิงวัตถุ ก็คือ กลุ่มของวัตถุที่ส่งข้อความข่าวสาร(Message) ถึงกันและกันเพื่อบอกให้วัตถุทำงาน

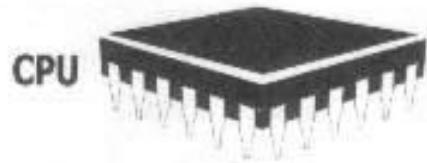


# วัตถุมีหน่วยความจำ

- วัตถุในความหมายของการเขียนโปรแกรมเชิงวัตถุ จะต้องมียหน่วยความจำของตัวเอง หรือนั่นคือส่วนที่เก็บข้อมูลส่วนของวัตถุนั้นๆ เองซึ่งเราเรียกว่า attribute
- ส่วนของข้อมูลนี้อาจจะประกอบขึ้นจากวัตถุอื่นๆ ได้
- เป็นที่มาของการเกิดคุณสมบัติการถ่ายทอดและคุณสมบัติขององค์ประกอบคอมพิวเตอร์

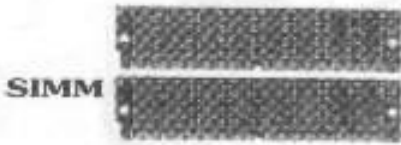


# หน่วยความจำคอมพิวเตอร์



CPU

Register

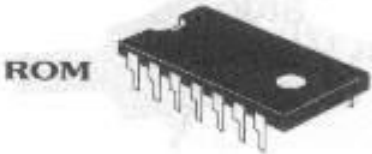


SIMM

Stack

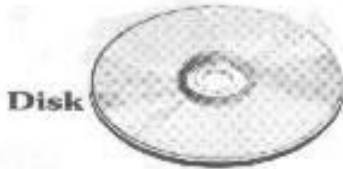
Heap

Fix location Storage



ROM

Control Storage



Disk

Non-RAM Storage

หน่วยความจำประเภท  
ต่าง ๆ และการเก็บข้อมูล

# วัตถุจะต้องจัดอยู่ในกลุ่มหรือชนิด

- ชนิดของวัตถุดังกล่าวก็คือคลาส (Class)
- ในการเขียนโปรแกรมเชิงวัตถุจะต้องเขียนคลาสขึ้นมาก่อนเสมอ แล้วจึงจะสร้างวัตถุจากคลาส ดังนั้นวัตถุจึงจะต้องอยู่ในคลาสใดคลาสหนึ่ง
- ดังนั้นคลาสเปรียบเหมือนแม่พิมพ์ ส่วนวัตถุก็คือสิ่งที่เกิดจากแม่พิมพ์ วัตถุที่เกิดจากคลาสดียวกันจึงมีคุณสมบัติพื้นฐานเหมือนกัน

# ตัวอย่างชนิดของวัตถุและวัตถุ

|             |  |
|-------------|--|
| MobilePhone |  |
|-------------|--|

|         |          |
|---------|----------|
| String  | bandname |
| Button  | button   |
| Screen  | screen   |
| Antenna | antenna  |
| Simm    | simNet   |

|           |
|-----------|
| call()    |
| receive() |
| hangup()  |
| sendSMS() |



**:siemens**



**:ericsson**



**:nokia**

# วัตถุประเภทเดียวกันย่อมได้ข่าวสารเหมือนกัน

- วัตถุที่อยู่ในประเภทเดียวกันย่อมได้ข่าวสารเหมือนกัน คุณสมบัติข้อนี้ทำให้การเขียนโปรแกรมเชิงวัตถุมีประสิทธิภาพเพิ่มขึ้นเพราะชนิดของวัตถุได้กำหนดหรืออยู่อย่างใด ๆ แต่สามารถจัดเป็นกลุ่ม ผู้เขียนคำสั่งสามารถกำหนดชนิดย่อย ๆ ของวัตถุได้อีก เช่น คลาสคือ shape เราสามารถแตกออกเป็นชนิดที่จำเพาะลงไปเป็น Circle, Triangle, Rectangle เป็นต้น ดังนั้นเมื่อวัตถุในคลาส Circle สามารถได้รับข่าวสารเหมือนกับ Triangle และ Rectangle

# แนวคิดที่เป็นประโยชน์จากการโปรแกรมเชิงวัตถุ

การโปรแกรมเชิงวัตถุ (Object-Oriented Programming) เป็นแนวคิดที่มีประโยชน์มากมายในการพัฒนาซอฟต์แวร์ ดังนี้:

1. การสร้างโครงสร้างและการจัดการข้อมูล: การโปรแกรมเชิงวัตถุช่วยให้เราสามารถสร้างโครงสร้างข้อมูลที่ต้องการและมีความยืดหยุ่น โดยใช้คลาสและออบเจกต์ เราสามารถจัดการข้อมูลได้อย่างมีระเบียบและเป็นระเบียบ และสามารถนำโครงสร้างที่มีอยู่แล้วมาใช้ซ้ำได้.
  2. การสืบทอด (Inheritance): การสืบทอดเป็นหนึ่งในคุณสมบัติสำคัญของโปรแกรมเชิงวัตถุ ช่วยให้เราสามารถสร้างคลาสใหม่โดยอิงจากคลาสที่มีอยู่แล้ว ซึ่งช่วยลดการเขียนโค้ดซ้ำซ้อน และเพิ่มความยืดหยุ่นในการปรับปรุงและการส่งต่อโค้ด.
  3. การแยกแอกสาร (Encapsulation): การโปรแกรมเชิงวัตถุช่วยให้เราสามารถแยกแอกสารได้ โดยปกป้องข้อมูลที่สำคัญและรายละเอียดเท่าที่จำเป็น ซึ่งช่วยเพิ่มความปลอดภัยและลดความซับซ้อนในการใช้งาน.
  4. พอลิมอร์ฟิสม์ (Polymorphism): คุณสมบัตินี้ช่วยให้เราสามารถใช้วัตถุของคลาสที่แตกต่างกันในลักษณะที่เดียวกัน โดยไม่จำเป็นต้องระบุตัวแปรใหม่ นี้ช่วยให้โค้ดสามารถทำงานกับวัตถุที่มีคลาสต่างๆ ได้อย่างยืดหยุ่น.
- ตัวอย่างเช่นในการพัฒนาแอปพลิเคชันการจองโรงแรม โปรแกรมเชิงวัตถุช่วยให้เราสามารถสร้างคลาสเช่น "โรงแรม" และ "ผู้ใช้งาน" และใช้คุณสมบัติการสืบทอดเพื่อสร้างคลาสย่อยที่มีคุณสมบัติเฉพาะ เช่น "โรงแรมหรู" หรือ "ผู้ใช้งานพิเศษ" ซึ่งช่วยลดการเขียนโค้ดซ้ำซ้อนและเพิ่มความยืดหยุ่นในการแก้ไขและปรับปรุงระบบ.
- การโปรแกรมเชิงวัตถุช่วยให้โค้ดมีความเป็นระเบียบ รูปแบบการพัฒนาที่ยืดหยุ่น และสามารถนำไปใช้ซ้ำในโปรเจกต์อื่นๆ นอกจากนี้ การใช้แนวคิดของการโปรแกรมเชิงวัตถุยังช่วยให้เราสามารถทำงานร่วมกันในทีมได้อย่างมีประสิทธิภาพ และช่วยลดความผิดพลาดในการพัฒนาโปรแกรมด้วย

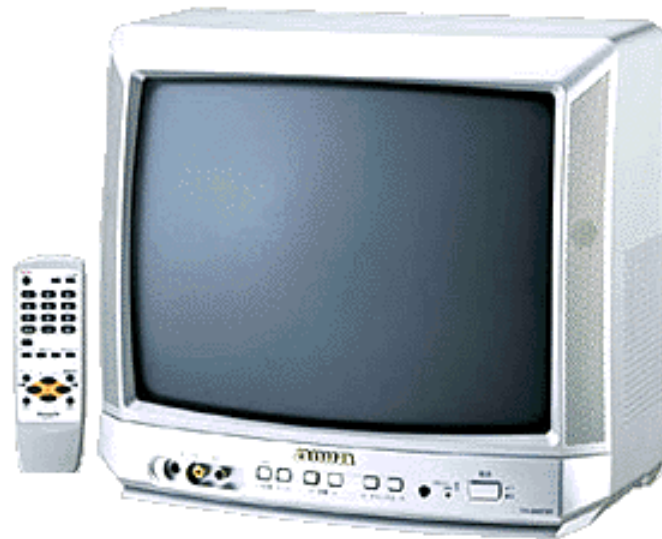
# แนวคิดที่เป็นประโยชน์จากการโปรแกรมเชิงวัตถุ

- การซ่อนรายละเอียด (information hiding)
- การเชื่อมต่อ (Interface)
- การนำวัตถุมาใช้ใหม่ (Reusable)
- การพ้องรูป: หนึ่งรูปหลายพฤติกรรม (Polymorphism)

# การซ่อนรายละเอียด(Information Hiding)

- มีผู้สังเกตเห็นว่าในชีวิตประจำวันของเรานั้น ไม่จำเป็นต้องทราบหรือรู้ทุกเรื่องก็สามารถใช้งานสิ่งของนั้นๆ ได้ เช่น

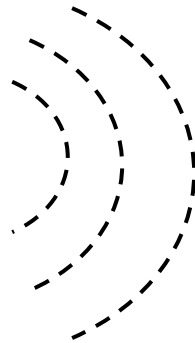
โทรทัศน์ที่มีอยู่ตามบ้าน เราไม่ต้องทราบการทำงานของอุปกรณ์ภายในของมัน แต่เราก็สามารถเปิดเครื่องและดูรายการโปรดของเราได้ เพียงรู้วิธีการกดปุ่มต่างๆ





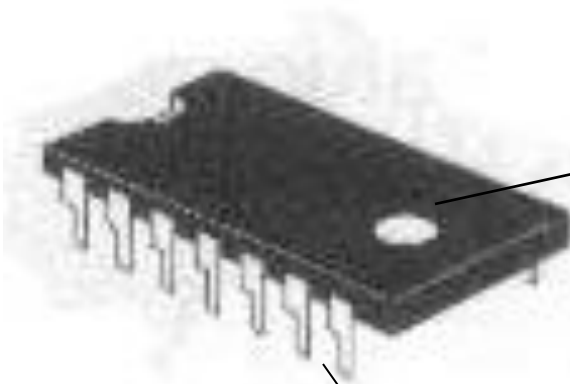
# การเชื่อมต่อ(Interface)

- โดยปรกติจะเชื่อมต่อระหว่างผู้ใช้กับคอมพิวเตอร์ เรียกว่า user interface



แต่คำว่าอินเตอร์เฟสสำหรับวัตถุคือ ส่วนที่บอกกว่าวัตถุนั้นให้บริการหรือสามารถทำอะไรได้บ้าง

# Information hiding



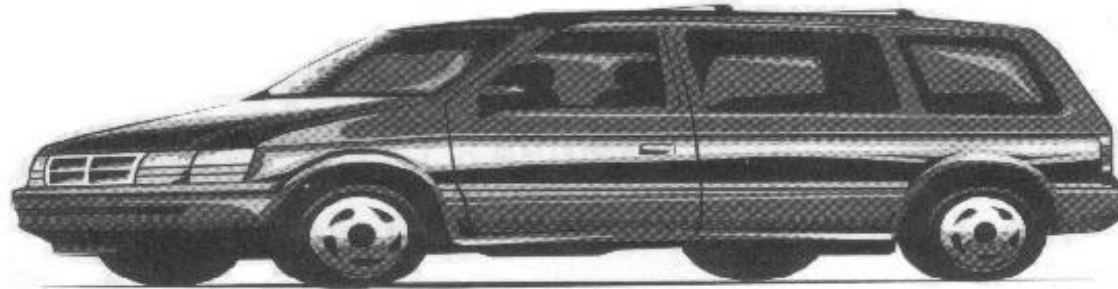
Information hiding  
เราจะไม่รู้การทำงานภายใน  
ของ ชิฟเพราะถูกซ่อน  
รายละเอียดไว้ แต่ยังสามารถ  
ใช้งานตามปกติ หากชิฟชิ้นนี้  
เสียก็สามารถหาตัวอื่นมา  
เสียบแทนได้เลย

ชิฟมีส่วนที่ใช้งานการติดต่อกับ  
อุปกรณ์ภายนอก interface

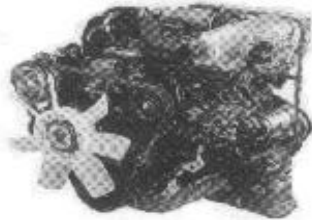
## การนำวัตถุมาใช้ใหม่(Reusable)

- จุดประสงค์ใหญ่ของการเขียนโปรแกรมเชิงวัตถุก็คือการนำส่วนต่างๆ ของวัตถุที่สร้างขึ้นกลับมาใช้ใหม่หรือที่เรียกในภาษาอังกฤษว่า “reuse” เมื่อมีวัตถุถูกสร้างขึ้นมา ก็สามารถนำวัตถุต่างๆ มาประกอบกันเป็นวัตถุอีกชนิดหนึ่งได้ เพื่อให้วัตถุใหม่มีความสามารถมากกว่าเดิม เช่นกรณีของรถยนต์สามารถประกอบขึ้นจาก เครื่องยนต์ ตัวถัง ประตู และล้อ เป็นต้น

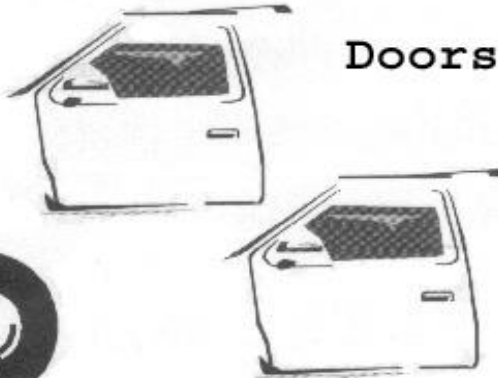
# ตัวอย่างวัตถุที่เกิดจากวิธีคอมโพสิต



**Engine**



**Doors**



**Wheels**



# การพ้องรูป (Polymorphism)

- รากฐานของการพ้องรูปคือคุณสมบัติการถ่ายทอด
- คุณสมบัติการถ่ายทอดยืนยันได้ว่าคลาสลูกที่เกิดจากคลาสแม่เดียวกันย่อมมีคุณสมบัติเหมือนกัน

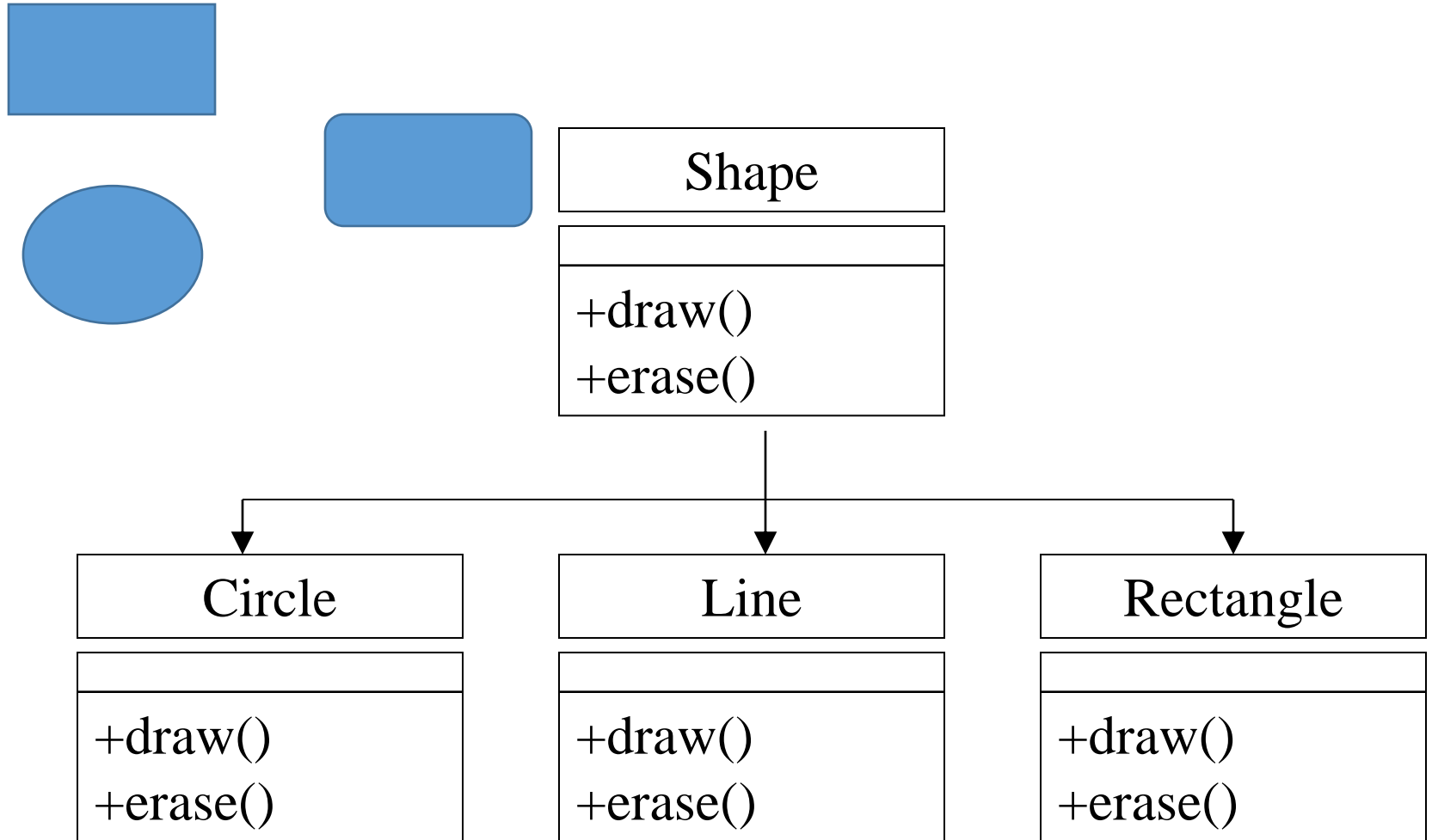
คลาสแม่คือ Shape

คลาสลูกคือ Circle, Triangle, Rectangle มีคุณสมบัติเหมือน

คลาสแม่ทุกประการ

- เป็นที่มาของหนึ่งรูปหลายพฤกษกรรม

# การถ่ายทอดให้เกิดลักษณะของพ็องรูป



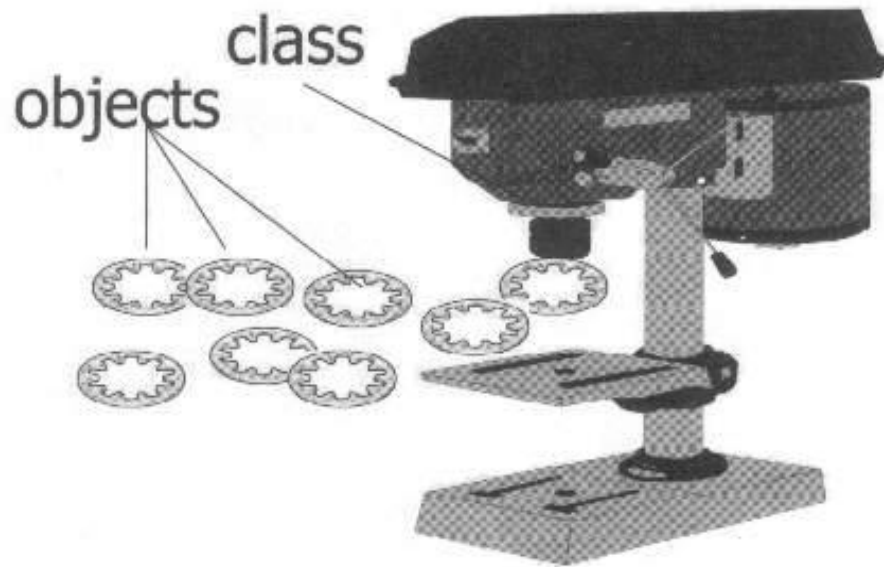
# มารู้จักคลาสและวัตถุกัน

- คลาสเป็นเหมือนส่วนของแม่พิมพ์ เพื่อบอกลักษณะว่าวัตถุในคลาสนี้มีลักษณะและพฤติกรรมเป็นอย่างไร
- วัตถุเป็นสิ่งที่ถูกสร้างขึ้นมาจากคลาสเพื่อใช้งาน กำหนดค่าต่างๆ ส่งข้อความสั่งให้วัตถุทำงาน

คลาสสี่เหลี่ยมผืนผ้า  
คุณสมบัติ: กว้าง, ยาว  
เมธอด: คำนวณพื้นที่

วัตถุA: สี่เหลี่ยมผืนผ้า  
กว้าง 2 ยาว 3  
เมธอด: คำนวณพื้นที่

# เปรียบเทียบคลาสกับวัตถุ



รูปแสดงความสัมพันธ์ระหว่างคลาสและวัตถุ



# วัตถุในโลกความจริง (Real-world objects)

วัตถุในโลกความจริง มีองค์ประกอบสำคัญสองสิ่งคือ

- สถานะ (State)
- พฤติกรรม (behavior)



สถานะ ชื่อ , สี , สายพันธุ์

พฤติกรรม เหา , วิ่ง , กระดิกหาง

สถานะ ตำแหน่งเกียร์ปัจจุบัน ,  
ความเร็วรอบปัจจุบัน , ความเร็วรถ

พฤติกรรม เปลี่ยนเกียร์ , เปลี่ยนความเร็วรอบ,  
เบรก





โคมไฟ

สถานะ ปิด , เปิด

พฤติกรรม กดปิด , กดเปิด



วิทยุ

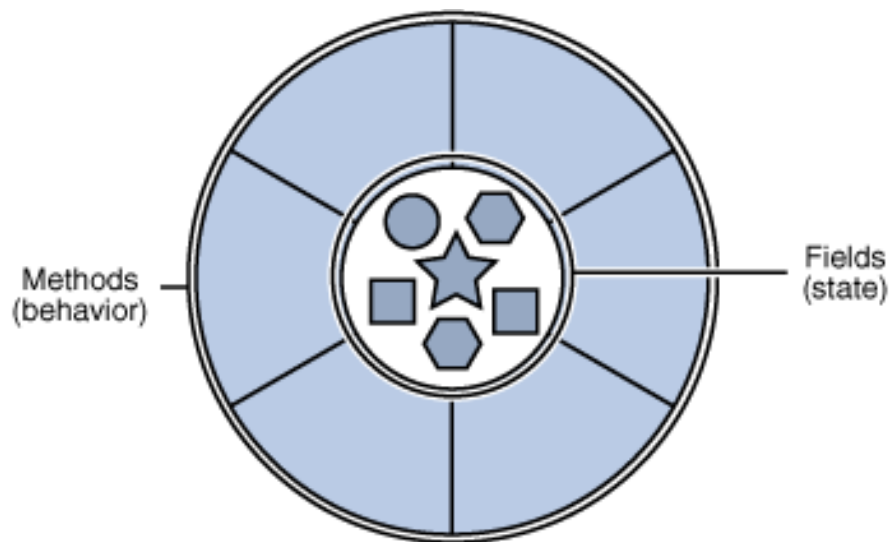
สถานะ ปิด, เปิด , ระดับเสียง, สถานีปัจจุบัน

พฤติกรรม กดปิด, กดเปิด , ลดเสียง , เร่งเสียง ,  
ค้นหาสัญญาณ

# วัตถุในโปรแกรม (Software objects)

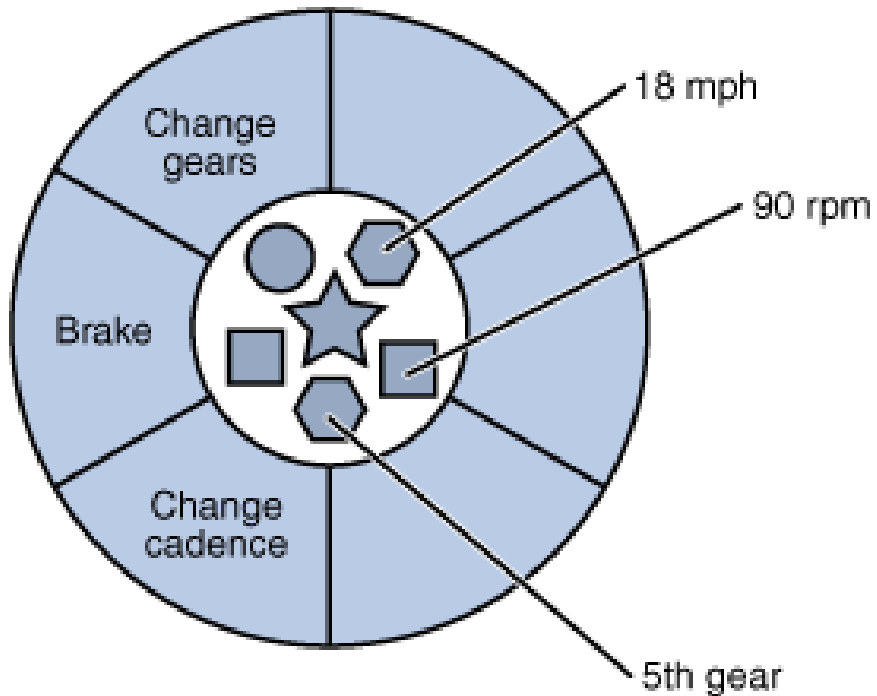
คล้ายคลึงกับวัตถุในโลกความจริง โดยจะประกอบไปด้วย สถานะ และพฤติกรรมเช่นเดียวกัน

- เก็บสถานะของวัตถุไว้ใน Field หรือ Attribute (ตัวแปรในโปรแกรม)
- และแสดงพฤติกรรมโดยผ่าน Method (ฟังก์ชันการทำงานในโปรแกรม)



# การทำงานระหว่าง Method กับ Field

- Method ทำหน้าที่เปลี่ยนแปลงสถานะ(Field)ซึ่งอยู่ในวัตถุนั้น ๆ และทำหน้าที่ติดต่อระหว่างวัตถุกับวัตถุ



Method Change gears ทำหน้าที่เปลี่ยนเกียร์

Method Brake ทำหน้าที่หยุดรถ

Method Change cadence ทำหน้าที่เปลี่ยนความเร็วรอบการปั่น

# ประโยชน์ของการใช้งาน Object ในโปรแกรม

- Modularity : โค้ดของโปรแกรมแบบ Object สามารถแยกส่วนกันเขียนได้อย่างเป็นอิสระ
- Information-hiding : รายละเอียดและข้อมูลถูกซ่อนอยู่ภายใน Object การปรับเปลี่ยนข้อมูลภายใน Object ต้องทำผ่าน Method เท่านั้น
- Code re-use : โค้ดของโปรแกรมสามารถนำกลับมาใช้ใหม่ได้อย่างสะดวก ทำให้พัฒนาโปรแกรมได้อย่างรวดเร็ว

## แบบฝึกหัดบทที่ 2

1. จงบอกประโยชน์ของการเขียนโปรแกรมเชิงวัตถุ
2. สถานะของออบเจกต์หมายความว่าอะไร?
3. จงจำแนกสิ่งต่อไปนี้ว่าสิ่งใดเป็นคลาส สิ่งใดเป็นออบเจกต์
  - สมุด, บ้านเลขที่ 97/8, ขนม, นายบัญชาพงศ์ ส่งเนียม , รถยนต์ , iPhoneX
4. จงหาข้อมูลและพฤติกรรมที่เป็นไปได้ของออบเจกต์ของคลาสต่อไปนี้
  - สุนัข, รถยนต์, โซฟา
5. จงแยกแยะว่าสิ่งใดเป็นซูปเปอร์คลาสและซับคลาส
  - ต้นไม้, ไม้ดอก
  - พนักงาน, ผู้จัดการ
  - เด็ก, ขนม
6. ให้นักศึกษายกตัวอย่างวัตถุในโลกความจริง 1 ตัวอย่าง แล้วแสดงให้เห็นว่าสถานะ (State) และ พฤติกรรม(Behavior) ของวัตถุนั้นมีอะไรบ้าง