



เขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



เขียนโปรแกรมเชิงวัตถุ

คือ การเขียนโปรแกรมอีกรูปแบบหนึ่ง โดยมองสิ่งต่างๆ เป็น
วัตถุ โดยในวัตถุจะมีคุณสมบัติและพฤติกรรมแนวคิดการเขียน
โปรแกรมมาจากพื้นฐานความจริงในชีวิตประจำวัน



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

องค์ประกอบพื้นฐาน



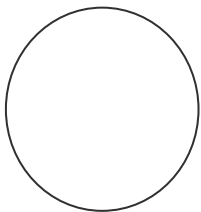
คลาส (class) คือ ต้นแบบของวัตถุการจะสร้างวัตถุขึ้นมาได้จะต้องสร้างคลาสขึ้นมาเป็นต้นแบบของวัตถุก่อนเสมอ

วัตถุหรือออบเจ็ค (object) คือ สิ่งที่ถูกสร้างจากคลาส
ประกอบด้วยคุณสมบัติ 2 ประการ คือ คุณลักษณะ และ พฤติกรรม

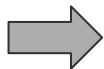
องค์ประกอบพื้นฐาน



Class



Animal



Object



สิงโต



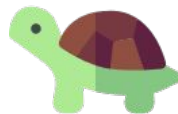
ช้าง



วัว



ไก่



เต่า



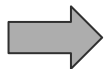
องค์ประกอบพื้นฐาน



Class



Employee



Accounting



Object



Programmer



Sale



<https://www.youtube.com/c/KongRuksiamOfficial/>

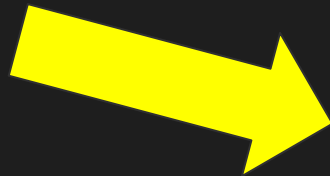
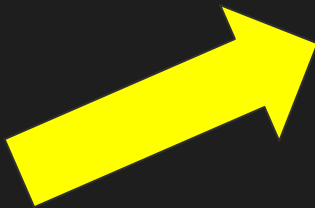
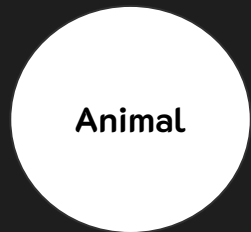


<https://www.facebook.com/KongRuksiamTutorial/>



องค์ประกอบพื้นฐาน

- **คุณลักษณะ (Field / Attribute)** สิ่งที่บ่งบอก
ลักษณะทั่วไปของวัตถุ
- **พฤติกรรม (Method)** คือ พฤติกรรมทั่วไปของวัตถุที่
สามารถกระทำได้



คุณสมบัติ (Field)

ชื่อ : ช้าง

สี : ฟ้าอ่อน

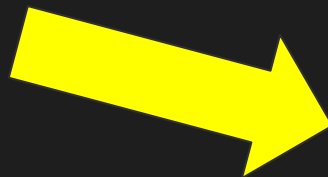
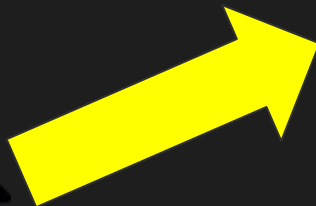
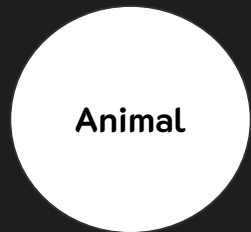
ประเภท : สัตว์บก

น้ำหนัก : 6 ตัน

จำนวนเท้า : 4 เท้า

พฤติกรรม (Method/Behavior)

- ร้อง
- นอน
- ส่งเสียงร้อง



คุณสมบัติ (Field)

ชื่อ : นก

สี : เหลือง

ประเภท : สัตว์ปีก

น้ำหนัก : 0.8 กิโลกรัม

จำนวนเท้า : 2 เท้า

พฤติกรรม (Method/Behavior)

- บิน
- เดิน
- ส่งเสียงร้อง



สรุปการเขียนโปรแกรมเชิงวัตถุ

- Class - ต้นแบบของวัตถุ
- Object - สิ่งที่ถูกสร้างขึ้นมาจาก Class ประกอบด้วย
 - คุณลักษณะ (Field)
 - พฤติกรรม (Method)
- คุณสมบัติของการเขียนโปรแกรมเชิงวัตถุ
 - การห่อหุ้ม (Encapsulation)
 - การสืบทอด (Inheritance)
 - การพ้องรูป (POLYMORPHISM)

การสร้าง Class & Object



การสร้าง Class

```
class class_name{
```

Field & Method

```
}
```

```
class Employee{
```

Field & Method

```
}
```



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



การสร้าง Object

โครงสร้างคำสั่ง

```
class_name obj_name;
```

ตัวอย่าง

```
Employee p1;
```



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



กฎการตั้งชื่อ

1. ชื่อ Class ควรกำหนดให้ตัวอักษรตัวแรกเป็นตัวพิมพ์ใหญ่ที่เหลือเป็นตัวพิมพ์เล็ก เช่น MyClass , Employee เป็นต้น
2. ชื่อ Object กำหนดเป็นตัวพิมพ์เล็กทั้งหมด
3. Field กำหนดเป็นตัวพิมพ์เล็ก เช่น name , age เป็นต้น

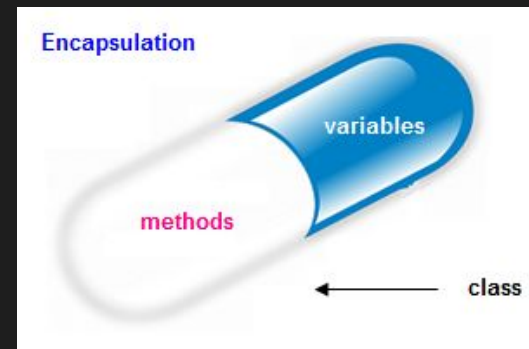


การห่อหุ้ม (Encapsulation)



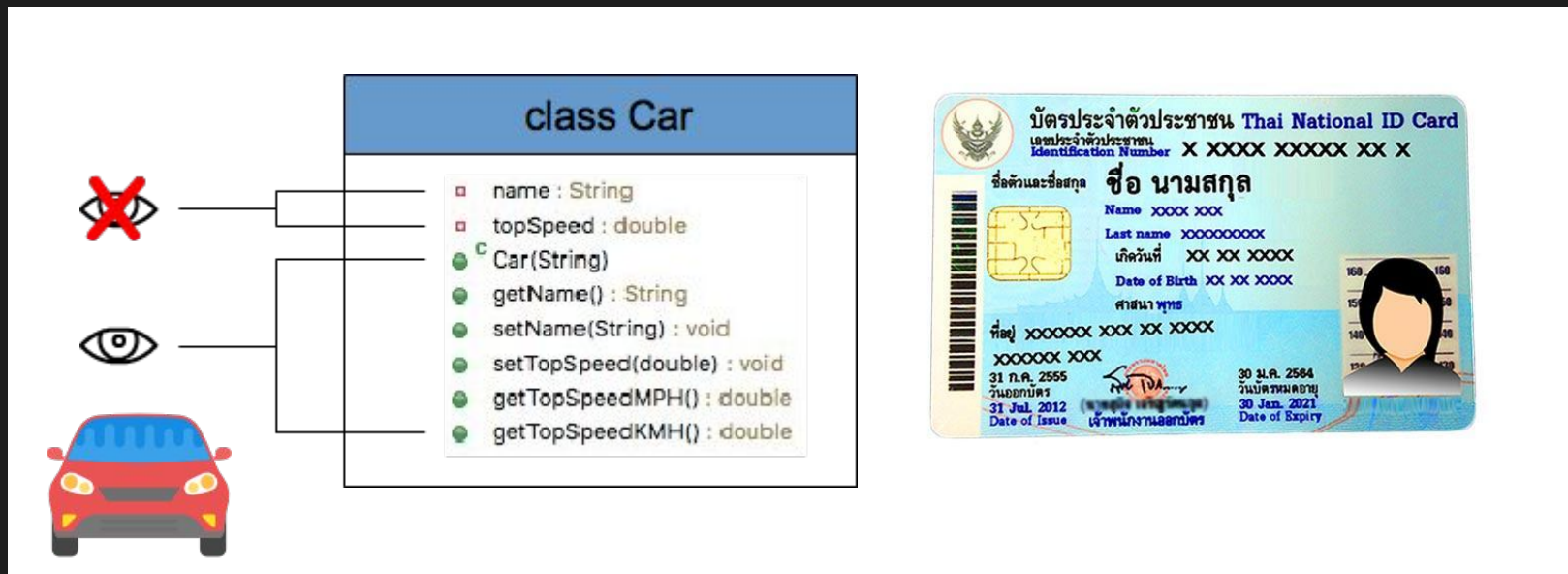
การห่อหุ้ม (Encapsulation)

1. เป็นกระบวนการซ่อนรายละเอียดการทำงานและข้อมูลไว้ภายใน ไม่ให้ภายนอกสามารถมองเห็นและสามารถทำการเปลี่ยนแปลงแก้ไขข้อมูลภายในได้ ซึ่งเป็นผลทำให้เกิดความเสียหายแก่ข้อมูล
2. สามารถสร้างความปลอดภัยให้แก่ข้อมูลได้ เนื่องจากข้อมูลจะถูกเข้าถึงจากผู้มีสิทธิ์เท่านั้น





การห่อหุ้ม (Encapsulation)





Access Modifier



Access Modifier

คือ ระดับในการเข้าถึง Class, Field, Method และอื่น ๆ ในภาษาเชิงวัตถุ มีประโยชน์อย่างมากในเรื่องของการกำหนดสิทธิ์ในการเข้าใช้งาน การซ่อนข้อมูลภายในคลาส และอื่น ๆ



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Access Modifier

- **Public** เป็นการประกาศระดับการเข้าถึงที่เข้มงวดน้อยที่สุดหรือกล่าวได้ว่าใครๆ ก็สามารถเข้าถึงและเรียกใช้งานได้
- **Protected** เป็นการประกาศระดับการเข้าถึงที่เกี่ยวข้องกับเรื่องการสืบทอด (Inheritance) ทำให้คลาสนั้นๆ สามารถเรียกใช้งานสมาชิกของคลาสที่ถูกกำหนดเป็น Protected ได้
- **Private** เป็นการประกาศระดับการเข้าถึงที่เข้มงวดที่สุด กล่าวคือ จะมีแต่คลาสของตัวเองเท่านั้นที่มีสิทธิ์ใช้งานได้





การสร้างฟิลด์ (Field)

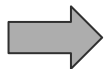


การสร้าง Field

Class



Employee



ชื่อ : เจน
เงินเดือน : xx,xxx

Object



ชื่อ : ก้อง
เงินเดือน : xx,xxx



ชื่อ : โจ้
เงินเดือน : xx,xxx





การสร้าง Field

โครงสร้างคำสั่ง

```
class className {  
    access_modifier :  
        type fieldName;  
}
```



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



ตัวอย่างการสร้าง Field

```
class Employee{  
    public :  
        field ที่กำหนดขอบเขตการเข้าถึงแบบ public  
  
    private :  
        field ที่กำหนดขอบเขตการเข้าถึงแบบ private  
  
}
```





การเรียกใช้งาน Field

เรียกใช้งานภายใน Class

- `fieldName`

เรียกใช้งานภายนอก Class

- `obj_name.fieldName`





การสร้างเมธอด (Method)



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



การสร้าง Method

โครงสร้างคำสั่ง

modifier :

```
return_type method_name (type parameter){
```

```
// statement
```

```
}
```

การเรียกใช้งาน

```
obj.method_name()
```



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Getter , Setter Method



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Constructor



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Constructor

เป็นเมธอดพิเศษที่จะถูกใช้เมื่อสร้างวัตถุและทำงานอัตโนมัติเพียงครั้งเดียวในตอนเริ่มต้นมีชื่อเหมือนกับชื่อคลาส

โครงสร้าง Constructor

```
className([parameter]){  
    //การทำงานด้านใน  
}
```



รูปแบบ Constructor

- Default Constructor คือ Constructor เริ่มต้นที่มีอยู่ในทุกคลาส
- Parameter Constructor คือ Constructor ที่สามารถส่งพารามิเตอร์เข้าไปทำงานได้



คุณสมบัติของ Constructor

- เป็นเมธอดที่ชื่อเหมือนคลาส
- ทำงานอัตโนมัติเพียงครั้งเดียวเมื่อตอนเริ่มต้นสร้าง Object ขึ้นมาใช้งาน
- ไม่มีการ return ส่งค่ากลับออกมา
- ไม่สามารถเรียกใช้งาน Object ด้วย Pointer ได้



Parameter Constructor

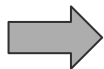


กำหนดค่าเริ่มต้น

Class



Employee



ชื่อ : เจน
เงินเดือน : xx,xxx

Object



ชื่อ : ก้อง
เงินเดือน : xx,xxx



ชื่อ : โจ้
เงินเดือน : xx,xxx





Constructor Overloading



Constructor Overloading

การสร้าง Constructor ที่มีชื่อเดียวกันแต่มีจำนวนพารามิเตอร์หรือประเภทข้อมูลที่แตกต่างกันภายในคลาส



การสืบทอดคุณสมบัติ (Inheritance)



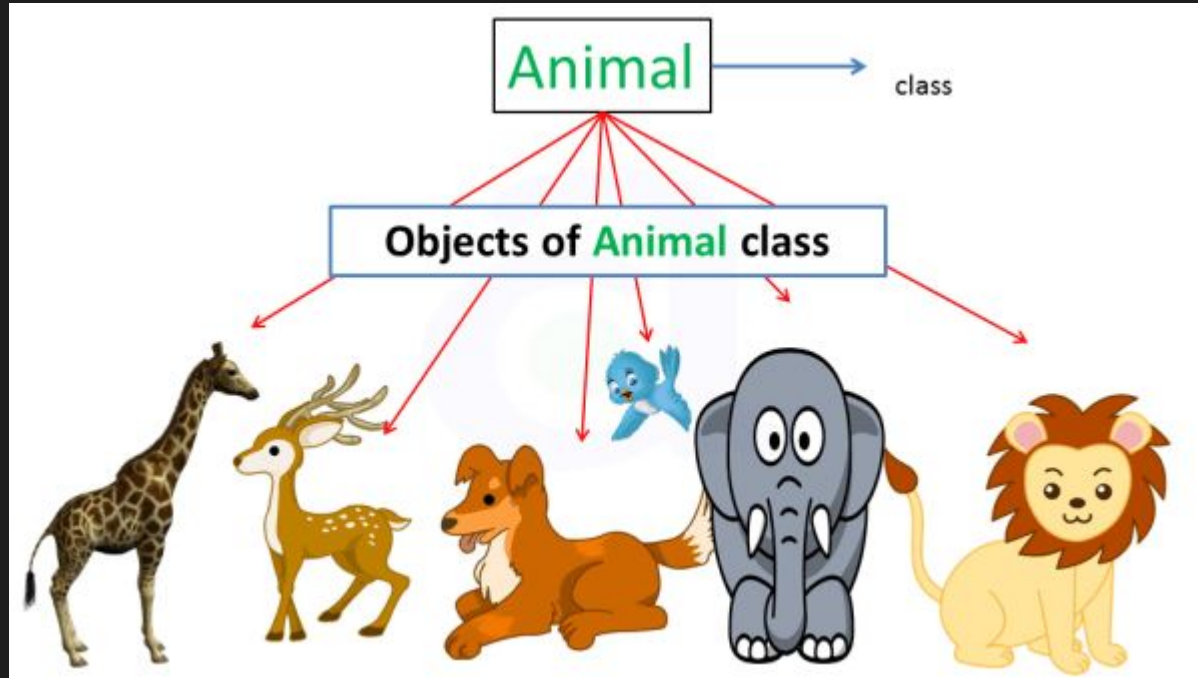
การสืบทอดคุณสมบัติ (Inheritance)

คือ การสร้างสิ่งใหม่ขึ้นด้วยการสืบทอด หรือรับเอา (inherit) คุณสมบัติบางอย่างมาจากสิ่งเดิมที่มีอยู่แล้ว

ข้อดีของการ inheritance คือ สามารถนำสิ่งที่เคยสร้างขึ้นแล้วนำกลับมาใช้ใหม่ (re-use) ได้ ทำให้ช่วยประหยัดเวลาการทำงานลง เนื่องจากไม่ต้องเสียเวลาพัฒนาใหม่หมด



คลาสแม่ (Base class) คลาสลูก (Derived Class)





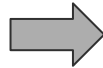
คุณสมบัติต่างๆจากแม่จะถูกถ่ายทอดไปยังลูก

Class

ยกเว้น Private Field และ Private Method



Employee



Accounting



Programmer



Sale



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



ชนิดของการสืบทอดคุณสมบัติ

- **การสืบทอดแบบ public** เป็นการสืบทอดที่ทำให้คลาสลูกรับการสืบทอดจากคลาสแม่มากที่สุด กล่าวคือ เมื่อกำหนดการสืบทอดเป็นรูปแบบ public คลาสลูกจะสามารถเข้าถึงสมาชิกในคลาสแม่ที่เป็นรูปแบบ public ได้ทั้งหมด (ยกเว้นสมาชิกแบบ private)
- **การสืบทอดแบบ private** เป็นการสืบทอดที่ทำให้สมาชิกแบบ public ที่ถูกสืบทอดมาจากคลาสแม่ เป็น private ส่วนสมาชิกที่เป็น private อยู่แล้วจะไม่สามารถเข้าถึงได้



การสืบทอดคุณสมบัติ (Inheritance)

คลาสแม่

```
class BaseClass{
```

```
// Field & Method
```

```
}
```

คลาสลูก

```
class DerivedClass : access BaseClass{
```

```
// Field & Method
```

```
}
```

access คือ ขอบเขตการเข้าถึงสมาชิกในคลาสแม่



ตัวอย่าง

คลาสแม่

```
class Employee{
```

```
// Field & Method
```

```
}
```

คลาสลูก

```
class Sale : public Employee{
```

```
// Field & Method
```

```
}
```



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Constructor Inheritance



Constructor Inheritance คืออะไร

เป็นการกำหนดให้ Class ลูกสามารถเรียกใช้งาน
Constructor ที่ทำงานอยู่ในคลาสแม่ได้

โครงสร้างคำสั่ง

```
DerivedConstructor() : ParentConstructor(){  
  
}
```



การพ้องรูป (POLYMORPHISM)



การพ้องรูป (POLYMORPHISM)

คุณสมบัติการพ้องรูป คือ สามารถตอบสนองต่อ Method เดียวกันด้วยวิธีการที่ต่างกันและสามารถกำหนด object ได้หลายรูปแบบมีข้อดี คือ ทำให้โปรแกรมสามารถปรับเปลี่ยนหรือเพิ่มเติมได้ง่ายขึ้น



<https://www.youtube.com/c/KongRuksiamOfficial/>



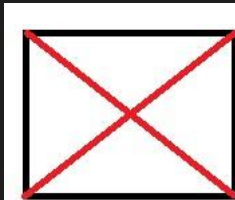
<https://www.facebook.com/KongRuksiamTutorial/>



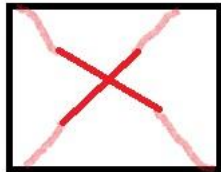
การพ้องรูป (POLYMORPHISM)

“ ข้อความเดียวกันแต่กระบวนการทำงานภายในแตกต่างกันนั้น
เรียกว่า การพ้องรูป หรือ polymorphism ”

กา



1



2





ดำ



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Overriding Method



Overriding Method

คือ เมธอดของคลาสลูก ที่มีชื่อเหมือนกับเมธอดของ
คลาสแม่ (เป็นผลมาจากคุณสมบัติ OO คือ inheritance)
แต่มีกระบวนการทำงานด้านในแตกต่างกัน



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Abstract Class



Abstract Class

คือ คลาสที่มีการสร้างเมธอดที่ไม่มีกระบวนการทำงานด้าน
ใน (Pure Virtual Method) หากคลาสใดสืบทอดมาจาก
Abstract class คลาสนั้นจะต้องทำการระบุเมธอดที่เป็น Pure
Virtual method ใน Class นั้นไว้เสมอ





ตัวอย่าง

```
class Employee{  
public:  
    virtual void method()=0; //pure virtual method  
}
```



Friend Method



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Friend Method คืออะไร

เป็นการประกาศคีย์เวิร์ด friend ไว้ด้านหน้าเมธอด ทำให้
ภายนอกสามารถเข้าถึงสมาชิกแบบ private ของคลาสผ่านเมธ
อดนี้ได้ โดยที่เมธอดดังกล่าวนั้นไม่ได้เป็นสมาชิกของคลาส



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Friend Method

โครงสร้างคำสั่ง

```
friend return_type method_name (type parameter){  
    // statement  
}
```



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Friend Class



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Friend Class คืออะไร

เป็นการประกาศคีย์เวิร์ด friend ไว้ในคลาส ทำให้
ภายนอกสามารถเข้าถึงสมาชิกแบบ private ของคลาสนั้นๆ
ได้หมายถึงคลาสที่เข้าถึงได้นั้นเป็นเพื่อนกับคลาสที่อ้างอิง



Static Field



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>



Static Field

คือ Field ที่สามารถเรียกใช้งานได้โดยตรง ไม่ต้องเรียกผ่าน Object
การสร้าง Static Field จะเหมือนกับการสร้าง Field โดยทั่วไปเพียงแค่เติม
static นำหน้า Field

โครงสร้างคำสั่ง

```
static type field = defaultvalue;
```

การเรียกใช้งาน

```
className::field
```