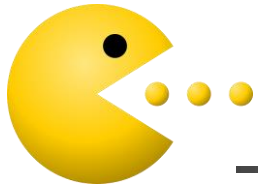


# Computer Game Programming

## การเขียนโปรแกรมคอมพิวเตอร์ แบบภาษาเชิงวัตถุ

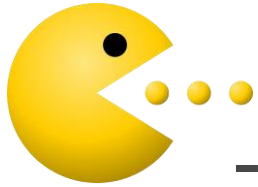
การเขียนโปรแกรมคอมพิวเตอร์แบบภาษาเชิงวัตถุ เป็นการเขียนโปรแกรมเพื่อให้สามารถใช้องค์ประกอบหรือการทำงานร่วมกัน แบ่งออกเป็นส่วนๆ สามารถใช้คุณสมบัติที่มีการพัฒนามาแล้ว เริ่มต้นผู้สนใจควรศึกษาถึงโครงสร้าง นิยาม ความหมายของภาษาเชิงวัตถุ



# ประกอบด้วย

---

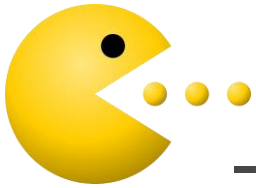
- พื้นฐานโปรแกรมคอมพิวเตอร์แบบภาษาเชิงวัตถุ
- ประโยชน์ของโปรแกรมคอมพิวเตอร์แบบภาษาเชิงวัตถุ
- การประยุกต์ใช้งานโปรแกรมเกมแบบภาษาเชิงวัตถุ



# พื้นฐานโปรแกรมคอมพิวเตอร์แบบ ภาษาเชิงวัตถุ

---

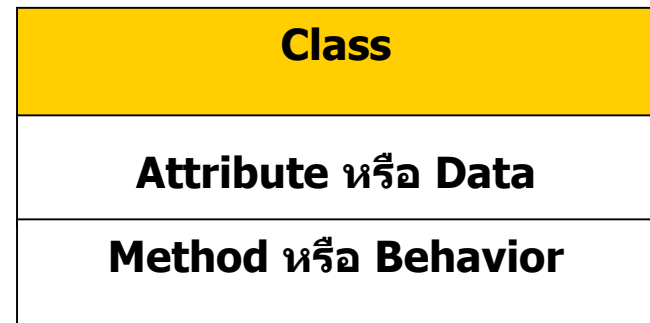
โปรแกรมแบบภาษาเชิงวัตถุ เป็นการนำ“วัตถุ” (object) มาประกอบกันเป็นโปรแกรมคอมพิวเตอร์ กำหนดให้องค์ประกอบในโปรแกรมคอมพิวเตอร์เป็นวัตถุ มีคุณสมบัติต่างๆ เพื่อใช้ในการประมวลผลและสามารถทำงานร่วมกับวัตถุอื่นๆ ได้ มีการสื่อสารกันได้

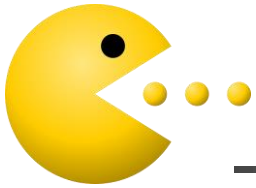


# คลาส(class)

---

คลาส(class) เปรียบเสมือนแม่พิมพ์บอกลักษณะว่าวัตถุในคลาสนี้มีลักษณะและพฤติกรรมเป็นอย่างไร ภายในคลาสประกอบด้วย แอททริบิวต์ (Attribute) หรือข้อมูล (Data) เป็นสิ่งที่ใช้บอกถึงลักษณะต่างๆของวัตถุในคลาสเมธอด(Method) หรือพฤติกรรม (Behavior) เป็นสิ่งที่ใช้อธิบายการทำงานของวัตถุในคลาส

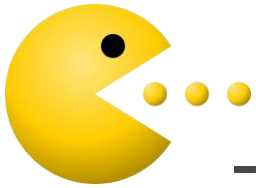




# คลาส(class)

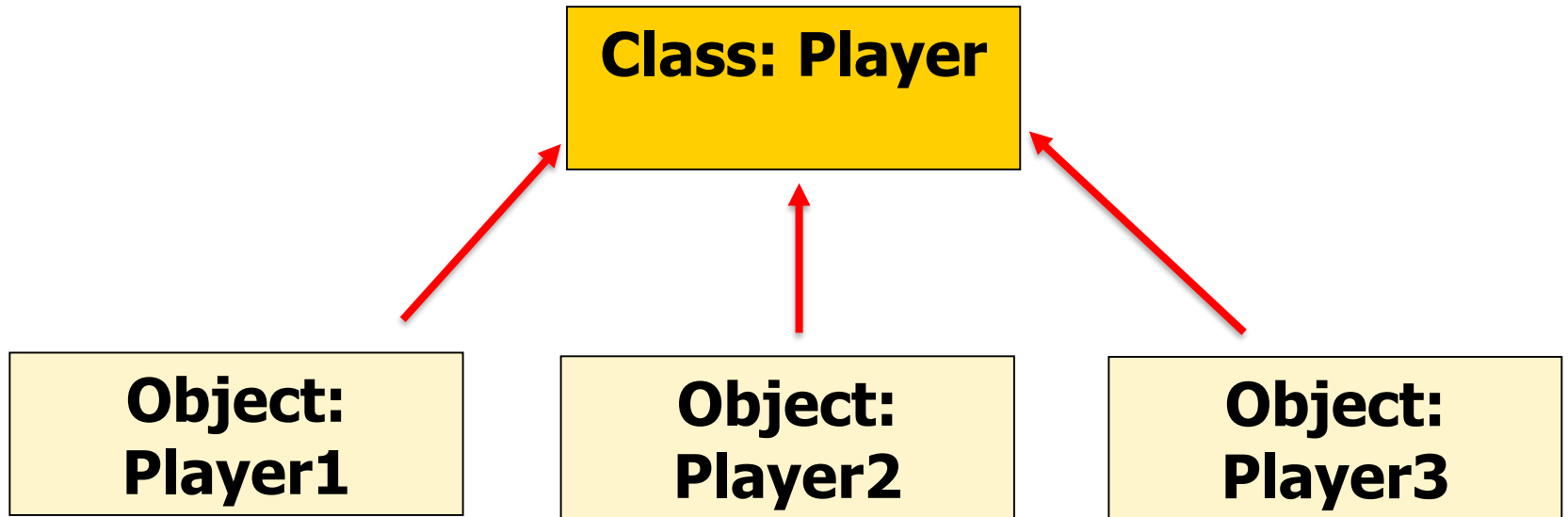
---

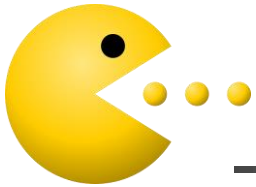
คลาสเปรียบเสมือนโปรแกรมที่เขียนขึ้นเพื่อทำหน้าที่เป็นพิมพ์เขียวของวัตถุ(object) การสร้างวัตถุจากคลาสเรียกว่าการทำ **instantiation** ผลลัพธ์ที่ได้คือ **instance** ของวัตถุ เรียก **object** หนึ่ง **object** ว่าเป็นหนึ่ง **instance** สามารถสร้าง **object** ได้หลายๆ **instance** จากคลาสเพียงคลาสเดียว ส่วนในภาษา **C#** หากสร้างคลาสหนึ่งคลาส ยกตัวอย่างเช่น สร้างคลาสขึ้นมาเพื่อนิยาม **node** หลังจากนั้น สามารถเขียนโปรแกรมคอมพิวเตอร์สร้าง **binary tree** ขณะทำงานอาจจะสร้าง **object** จากคลาส **node** ได้หลายล้าน **instance** ภายในหนึ่งวินาที เป็นต้น



# คลาส(class)

---





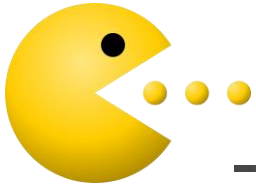
# คลาส(class)

การใช้งานคลาสในโปรแกรมคอมพิวเตอร์แบบ  
ภาษาเชิงวัตถุ สามารถทำได้ดังนี้คือ

- 1) ทำการสร้าง **object** และใช้งานคลาสผ่าน  
**object**
- 2) การเรียกใช้งานโดยตรงโดยไม่ต้องสร้าง  
**object** เรียกวิธีนี้ว่า **static class**

โครงสร้างรูปแบบของการสร้างคลาส

```
Class ชื่อคลาส  
{  
    method1(){  
        //วิธีการทำงาน  
    }  
    method2(){  
        //วิธีการทำงาน  
    }  
    attribute1;  
    attribute2;  
}
```



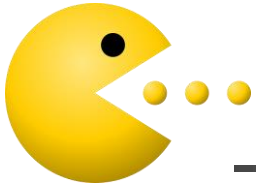
# คลาส(class)

---

วิธีการเข้าถึงหรือใช้ตัวแปร (**Variable**) หรือ ฟังก์ชัน (**Function**) สามารถประกาศการเข้าถึง ได้ดังนี้

- **public** คือ เมื่อประกาศตัวแปรหรือฟังก์ชันจะสามารถถูกอ้างถึงได้ทั้งหมด ทั้งภายในหรือภายนอกคลาส
- **protected** คือ จะถูกอ้างถึงได้เฉพาะในคลาสเดียวกันหรือคลาสลูก(คลาสที่มีการขยาย **extend** หรือสืบทอด **inherit** มาจากคลาสอื่น)
- **private** คือ เมื่อประกาศตัวแปรหรือฟังก์ชันจะสามารถใช้อ้างได้ภายในเฉพาะคลาสตัวเองเท่านั้น
- **static** จะถูกอ้างถึงได้โดยมีเงื่อนไขดังนี้ เมธอดที่เป็น **static** เช่น คลาส **in** สามารถเรียกได้เฉพาะเมธอดในคลาสเดียวกันที่เป็น **static** เท่านั้น เมธอดที่เป็น **static** จะถูกเรียกจากที่อื่นโดยอ้างผ่านทางชื่อคลาสได้ทันที ซึ่งต่างกับเมธอดตรงที่ไม่เป็น**static**ที่ต้องมีการสร้าง **instance** ก่อน





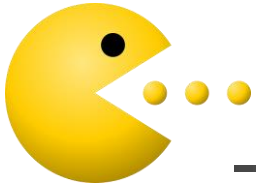
# ตัวอย่างโปรแกรมเกมคอมพิวเตอร์ ในการกำหนดการเข้าถึง

## ScopeAndAccessModifiers

```
using UnityEngine;
using System.Collections;

public class ScopeAndAccessModifiers : MonoBehaviour
{
    public int alpha = 5;
    private int beta = 0;
    private int gam = 5;
    private AnotherClass myOtherClass;

    void Start ()
    {
        alpha = 29;
        myOtherClass = new AnotherClass();
        myOtherClass.Fruitchine(alpha, myOtherClass.apples);
    }
}
```

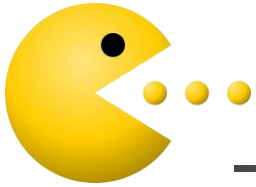


# ตัวอย่างโปรแกรมเกมคอมพิวเตอร์ ในการกำหนดการเข้าถึง

ScopeAndAccessModifiers

```
void Example (int pens, int crayons)
{
    int answer;
    answer = pens * crayons * alpha;
    Debug.Log(answer);
}

void Update ()
{
    Debug.Log("Alpha is set to: " + alpha);
}
}
```



# ตัวอย่างโปรแกรมเกมคอมพิวเตอร์ ในการกำหนดการเข้าถึง

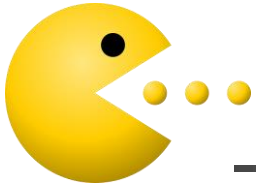
AnotherClass

```
using UnityEngine;
using System.Collections;

public class AnotherClass
{
    public int apples;
    public int bananas;

    private int stapler;
    private int sellotape;

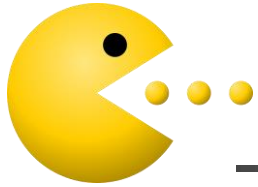
    public void FruitMachine (int a, int b)
    {
        int answer;
        answer = a + b;
        Debug.Log("Fruit total: " + answer);
    }
}
```



# ตัวอย่างโปรแกรมเกมคอมพิวเตอร์ ในการกำหนดการเข้าถึง

AnotherClass

```
private void OfficeSort (int a, int b)
{
    int answer;
    answer = a + b;
    Debug.Log("Office Supplies total: " + answer);
}
}
```



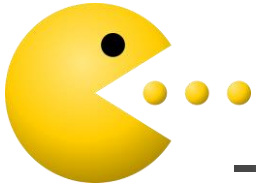
# ประโยชน์ของโปรแกรม คอมพิวเตอร์แบบภาษาเชิงวัตถุ

---

โปรแกรมคอมพิวเตอร์แบบภาษาเชิงวัตถุ มีประโยชน์ในการนำวัตถุ(object) ที่อยู่ในโปรแกรมคอมพิวเตอร์ที่พัฒนาแบบภาษาเชิงวัตถุ สามารถนำมาใช้ร่วมกันได้ สามารถทำการสืบทอดคลาส เป็นการสร้างสิ่งใหม่ขึ้นด้วยการสืบทอดหรือรับเอาคุณสมบัติบางอย่างมาจากสิ่งเดิมที่มีอยู่แล้วและสร้างเพิ่มเติมอีกได้ ทำให้ผู้พัฒนาโปรแกรมคอมพิวเตอร์ไม่ต้องเริ่มใหม่ทั้งหมด

การสร้างคลาสใหม่โดยการสืบทอดมาจากคลาสเก่า มีวิธีการดังนี้

- 1) สร้างคลาสใหม่ซึ่งสืบทอดมาจากคลาสเก่า เรียก คลาสใหม่ ว่าเป็น **ซับคลาส ของ คลาสเก่า** และ เรียก คลาสเก่า ว่าเป็น **ซูปเปอร์คลาส ของคลาสใหม่**
- 2) วัตถุของคลาสใหม่ มาจากคลาสเก่า **แต่วัตถุทั้งหมดของคลาสเก่าไม่จำเป็นต้องเป็นของ คลาสใหม่**

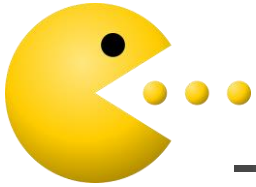


# ตัวอย่างโปรแกรมคอมไพเตอร์แสดง วิธีการสืบทอดคลาสที่มีชื่อว่า nager มา จากคลาสที่มีชื่อว่า Employee

---

```
public class nager : Employee
{
    // Employee fields, properties, methods and events are inherited

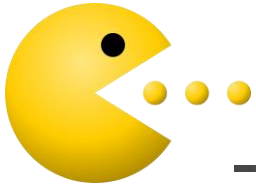
    // New nager fields, properties, methods and events go here...
}
```



# ประโยชน์อื่นๆ ของการเขียนโปรแกรม คอมพิวเตอร์แบบภาษาเชิงวัตถุ

---

- 1) การลดระยะเวลาในการพัฒนาโปรแกรมคอมพิวเตอร์โดยสามารถนำโค้ดโปรแกรมที่พัฒนาไว้กลับมาใช้งานร่วมกัน
- 2) การนำเอาโปรแกรมคอมพิวเตอร์ที่พัฒนาอยู่มาใช้งาน(reuse-ability)
- 3) การพัฒนาต่อยอดจากวัตถุ(object) ที่มีอยู่แล้ว
- 4) การดูแลแก้ไขทำได้ง่าย สามารถทำการแก้ไขเฉพาะ object ที่ต้องการเท่านั้น จะไม่กระทบต่อส่วนอื่นๆ ของโปรแกรมที่พัฒนาขึ้น
- 5) การพัฒนาแบบภาษาเชิงวัตถุ มีความเหมาะสมสำหรับระบบที่มีความซับซ้อน
- 6) การพัฒนาแบบภาษาเชิงวัตถุสามารถใช้คำหรือชื่อ ต่างๆ ที่มีอยู่ในระบบงานมาใช้ในการออกแบบโปรแกรม ทำให้เกิดความเข้าใจง่ายขึ้น
- 7) การพัฒนาแบบภาษาเชิงวัตถุมีความยืดหยุ่น สามารถปรับเปลี่ยนการทำงานของเดิมที่มีอยู่แล้วได้โดยไม่ต้องแก้ไขโปรแกรมคอมพิวเตอร์เดิม



## ใบงาน

ให้นักศึกษาแบ่งกลุ่มละ 2-3 คน อธิบายการพัฒนาโปรแกรมคอมพิวเตอร์แบบภาษาเชิงวัตถุ พร้อมยกตัวอย่างประกอบ และนำเสนอหน้าชั้นเรียน