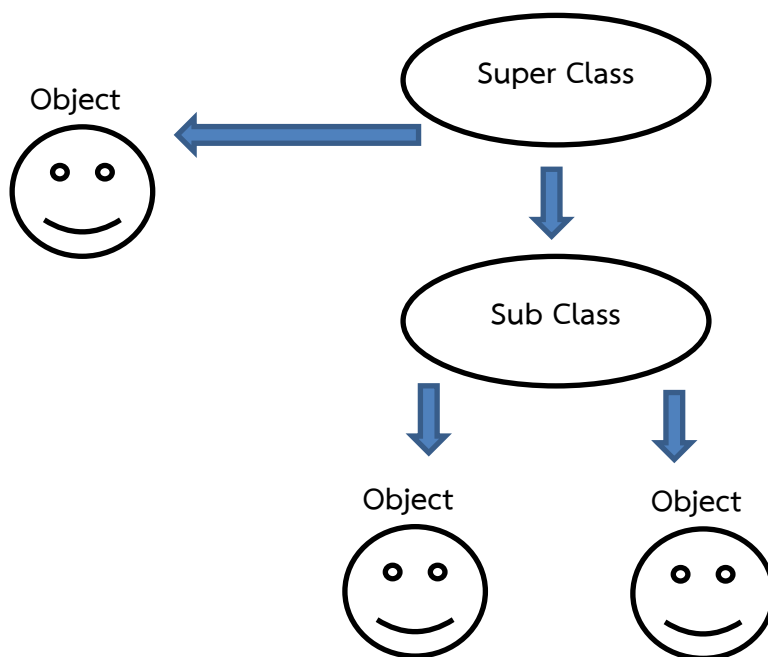


## บทที่ 5

### การเข้าถึงคลาส (Visibility)

จากที่กล่าวมาในเบื้องต้น อ็อบเจกต์จะเกิดจากคลาส แต่ในความเป็นจริงนั้น กว่าจะมีการสร้างอ็อบเจกต์ขึ้นมาใช้ได้ อาจเกิดจากคลาสที่ถูกแยกออกมาจากคลาสอื่นก็ได้ ดังโครงสร้างรูปที่ 5.1



รูปที่ 5.1 การถ่ายทอดจากคลาสไปอ็อบเจกต์

จากตัวอย่างที่ 5.1 แสดงลักษณะโครงสร้างของทำงานแบบเชิงวัตถุ ในกรณีแรกอาจมีคลาสเดียวและแยกออกมาเป็นอ็อบเจกต์ได้ หรืออีกกรณีอาจเกิดจากคลาสที่อยู่บนสุดจะเรียกว่า “คลาสหลัก” (Super Class) และมีการแยกออกมาเป็นอีกคลาสหนึ่ง เรียกว่า “คลาสย่อย” (Sub Class) ก่อนที่จะแยกออกเป็นอ็อบเจกต์

ดังนั้นจากกรณีที่กล่าวมา จะทำให้เกิดกรณีการใช้งานคลาสนั้น ซึ่งจะมีวิธีการเข้าถึงคลาสหลายวิธีการ แต่ละวิธีการจะมีการกำหนดสิทธิ์การใช้งานคลาสดังกล่าวแตกต่างกันออกไป ซึ่งสามารถแยกได้เป็น 3 ประเภท Public Private และ Protected ซึ่งการเข้าใช้แต่ละประเภท จะถูกกำหนด

ในส่วนของ แอททริบิวต์ และเมธอดของคลาส สำหรับรูปแบบของคลาสและออบเจ็ค การทำงานแบบเชิงวัตถุ แสดงดังรูปที่ 5.1

### 5.1 Public

การเข้าใช้งานคลาสลักษณะแบบ public นี้ เมื่อมีการสร้างอ็อบเจ็คต์จากคลาส อ็อบเจ็คต์นั้นจะสามารถเข้าถึง และเรียกใช้ แอททริบิวต์และเมธอด ของคลาสนั้นได้โดยทันที ดังตัวอย่างโปรแกรมที่ 5.1 แสดงการเข้าถึงคลาสแบบ public

ตารางที่ 5.1

บรรทัดที่	ชุดคำสั่ง
1	<?
2	class mother
3	{
4	public \$name="";
5	var \$sex="หญิง";
6	public \$height=165;
7	public \$weight=60;
8	public \$age=30;
9	
10	public function run()
11	{
12	echo "you can running";
13	}
14	public function jump()
15	{
16	echo "you can jumping";
17	}
18	}
19	
20	\$somchat = new mother;
21	echo "ชื่อ = ".\$somchat->name." ";

22	echo "เพศ = ".\$somchat->sex." ;
23	echo "สูง = ".\$somchat->height." ;
24	echo "กว้าง = ".\$somchat->weight." ;
25	echo "อายุ = ".\$somchat->age." ;
26	
27	\$somchat -> run();
28	echo " ;
29	\$somchat -> jump();
30	
31	?>

#### ผลลัพธ์

ชื่อ = กาญจนา การรัมย์
เพศ = หญิง
สูง = 165
กว้าง = 60
อายุ = 30
you can running
you can jumping

#### คำอธิบาย

3.1.1 ในการกำหนดลักษณะของ Public นั้น จะประกาศโดยใช้คำว่า “Public” หน้า แอททริบิวต์ และเมธอด หรือไม่ก็ได้ ซึ่งถ้าไม่ได้กำหนดอะไร โปรแกรมจะเข้าใจว่า คุณสมบัติแบบนั้นเป็น Public

3.1.2 คุณสมบัติแบบ Public นั้นทั้งคลาสหรืออ็อบเจกต์ สามารถเรียกใช้แอททริบิวต์และเมธอดแบบนี้ได้ทันที และค่าต่างๆ ของคุณสมบัติแบบนี้ เมื่อมีการเปลี่ยนแปลงค่า จะมีผลต่อค่าแอททริบิวต์ในคลาส ดังในบรรทัดที่ 21-29

## 5.2 Private

ในการประกาศการเข้าถึงคลาสแบบนี้ จะประกาศใช้งานเฉพาะในคลาสเท่านั้น จะไม่ได้รับถ่ายทอดไปยังอ็อบเจกต์

บรรทัดที่	ตัวอย่างที่ 5.2 (ให้เปลี่ยนคำสั่ง ตามบรรทัดข้างล่างนี้)
8	private \$age=30;

ผลลัพธ์

ชื่อ = กาญจนา การรัมย์

เพศ = หญิง

สูง = 165

กว้าง = 60

**Fatal error:** Cannot access private property human::\$age in  
C:\AppServ\www\OOP\02\_Private.php on line 25

### คำอธิบาย (ตัวอย่างที่ 5.2)

5.2.1 เมื่อมีการประกาศคุณสมบัติ เป็น private ในคลาสแล้ว ออบเจ็ค จะไม่สามารถเข้าใช้งานคลาสได้ เมื่อมีการเรียกใช้ จะเกิด error ของโปรแกรม

5.2.2 การใช้งานแบบ private จะประกาศเพื่อใช้งานภายในคลาสนั้น ดังตัวอย่างที่ 5.3

บรรทัดที่	ตัวอย่างที่ 5.3
1	<?
2	class human
3	{
4	public \$name="กาญจนา การรัมย์";
5	var \$sex="หญิง";
6	public \$height=165;
7	public \$weight=60;
8	private \$age;
9	
10	public function sayhi(){
11	echo "สวัสดี "." ".\$this->name." คุณอายุ = ".\$this->setAge();
12	}
13	private function setAge()
14	{
15	return \$this->age="35";
16	}
17	}

18	\$obj = new human;
19	\$obj -> sayhi();
20	?>

ผลลัพธ์

สวัสดี กาญจนา การรัมย์ คุณอายุ =35

คำอธิบาย (ตัวอย่างที่ 5.3)

ลักษณะการประกาศแบบ private ก็เพื่อนำไปใช้ในการประมวลผล หรือกำหนดทำงานภายในคลาสเท่านั้น ดังบรรทัดที่ 8-16

### 5.3 Protected

เป็นการกำหนดการเข้าถึงคลาส ภายในคลาสของตนเอง และภายในคลาสลูก (Sub Class) โดยใช้คำ protected ดังตัวอย่างที่ 4.4

บรรทัดที่	ตัวอย่างที่ 4.4
1	<?
2	class mother
3	{
4	public \$name="กาญจนา การรัมย์";
5	protected \$sex="หญิง";
6	public \$height=165;
7	public \$weight=60;
8	private \$age="30";
9	
10	public function sayhi(){
11	echo "สวัสดี "." ".\$this->name." คุณอายุ = ".\$this->setAge();
12	}
13	private function setAge()
14	{
15	return \$this->age="35";
16	}

17	}
18	
19	class son extends mother
20	{
22	public function showsex()
23	{
24	return \$this->sex;
25	}
26	public function showage()
27	{
28	return \$this->age;
29	}
30	}
31	
32	\$somchat = new son;
33	echo "เพศ ".\$somchat->showsex();
34	echo " ";
35	echo "อายุ ".\$somchat->showage();
36	
37	?>
38	

### คำอธิบาย

1. เมื่อกำหนดคุณสมบัติเป็น Protected แล้ว ในบรรทัดที่ 6
2. สร้างคลาสตัวที่ woman จากคลาส human (ใช้ คีย์เวิร์ด extends) ในบรรทัดที่ 20
3. สร้างออบเจ็กต์จากคลาส woman ในบรรทัดที่ 33
4. เรียกใช้ผ่านเมธอด showsex() และ showage()

### ผลลัพธ์

เพศ หญิง

อายุ

จากผลลัพธ์จะเห็นได้ว่า คุณสมบัติ sex ที่อยู่คลาส human ได้รับการถ่ายทอดมายัง woman ซึ่งผลลัพธ์ก็คือ เพศ หญิง แต่ในส่วนของคุณสมบัติ age ที่อยู่ในคลาส human ไม่ได้รับการถ่ายทอดมายัง woman นี้ข้อแตกต่างระหว่าง protected และ private

สรุปคุณสมบัติแบบ Protected จะเป็นคุณสมบัติที่เกิดจากคลาสแม่ และคลาสลูกสามารถเรียกใช้คุณสมบัตินี้ได้ แต่อ็อบเจกต์ที่เกิดจากคลาส ไม่สามารถเรียกใช้คุณสมบัตินี้โดยตรงได้ จะเรียกใช้คุณสมบัติผ่านเมธอดของคลาสลูกเท่านั้น

อธิบายคำสั่งท้ายบท	
คำสั่ง	ความหมาย
\$this	ตัวอย่างอิงไปยัง คุณสมบัติ หรือ เมธอดของคลาส
->	ตัวโอเปอเรเตอร์ (ตัวเครื่องหมาย) ที่ใช้อ้างอิงไปยัง คุณสมบัติ หรือ เมธอด
return	การคืนค่า ซึ่งจะประกาศใช้ภายในเมธอด