



ความสัมพันธ์ระหว่างคลาส (Class Relationships)

**Matinee Kiewkanya
Computer Science, CMU**

ความสัมพันธ์ระหว่างคลาส

Class Relationships คือความสัมพันธ์ระหว่างคลาส
ที่มีการทำงานร่วมกัน โดยมีประเภทของ
ความสัมพันธ์หลายประเภทดังนี้

❁ **Generalization**

❁ **Association**

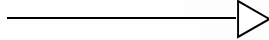
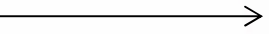
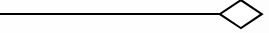
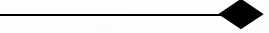
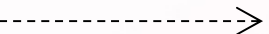
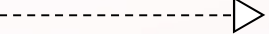
❁ **Aggregation**

❁ **Composition**

❁ **Dependency**

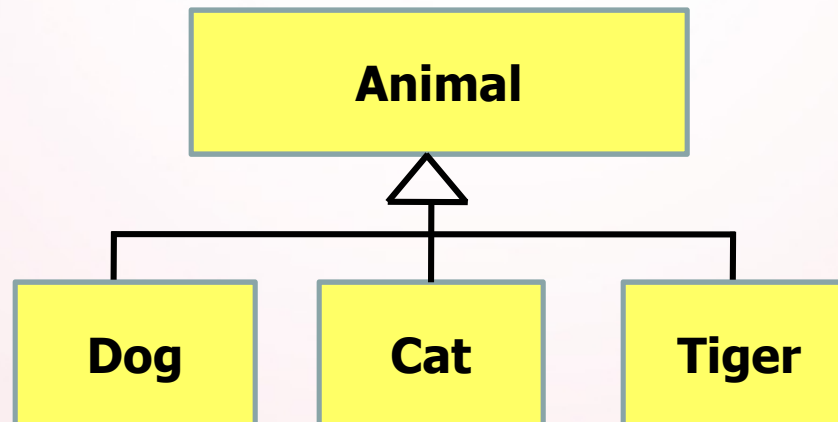
❁ **Realization**

ความสัมพันธ์ระหว่างคลาส

ความสัมพันธ์	สัญลักษณ์
Generalization	
Association	
Aggregation	
Composition	
Dependency	
Realization	

Generalization

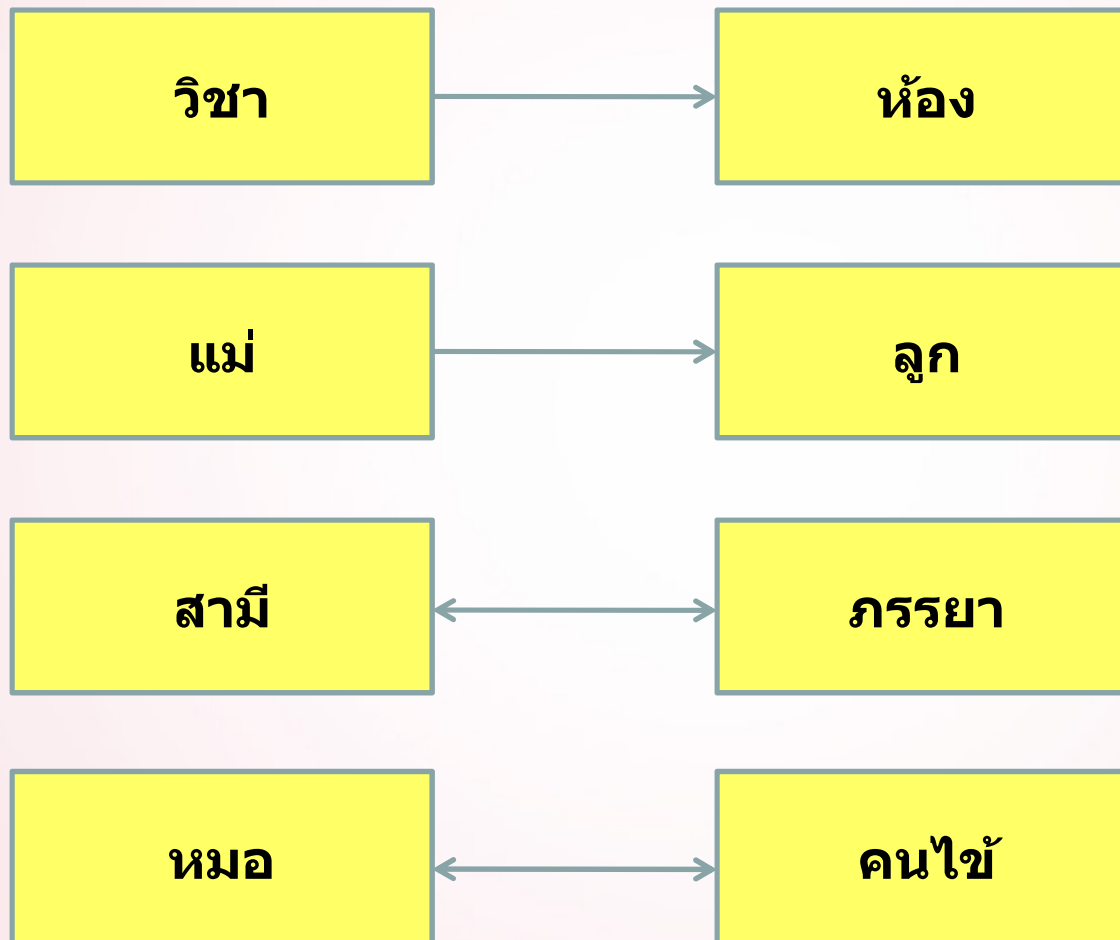
- ❁ เป็นความสัมพันธ์แบบสืบทอดคุณสมบัติ (Inheritance)
- ❁ มีการสร้างคลาสใหม่ขึ้นมาจากคลาสพื้นฐาน คลาสใหม่นี้จะมีลักษณะพื้นฐานเหมือนกับคลาสเดิม แต่จะมีลักษณะพิเศษที่เพิ่มเติมขึ้นมา ซึ่งจะมีความเฉพาะเจาะจงมากยิ่งขึ้น เป็นความสัมพันธ์แบบ "is a"



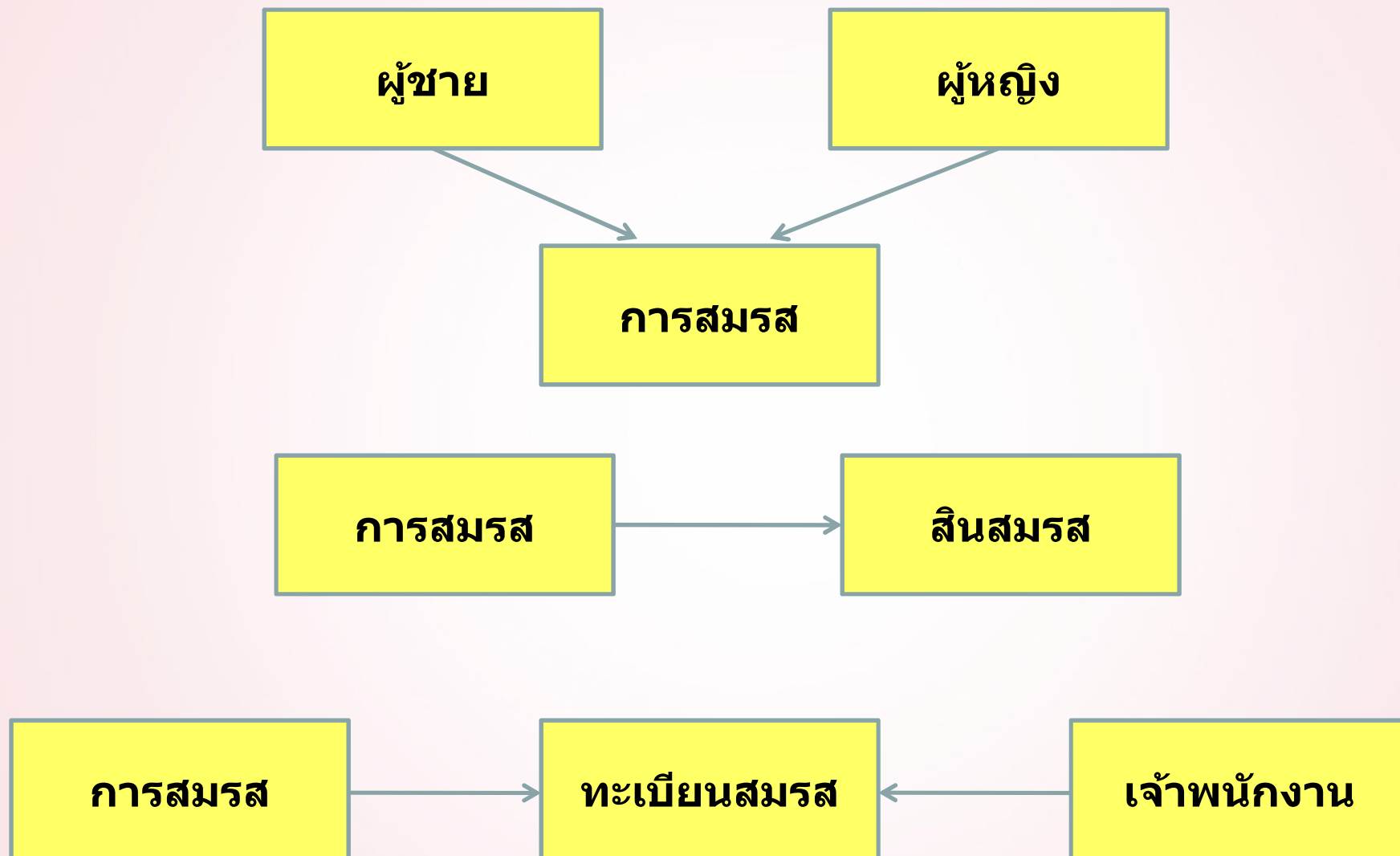
Association

- ❁ เป็นความสัมพันธ์แบบเกี่ยวข้อต่อกัน หรือ “is related to”
- ❁ ความสัมพันธ์แบบนี้อาจสร้างขึ้นได้โดยการกำหนด แอททริบิวต์ของคลาสให้มีชนิดเป็นคลาสอีกคลาสหนึ่ง และสามารถเรียกใช้เมทอดจากคลาสนั้นได้ในกรณีที่ต้องการ
- ❁ Association จะมีทั้งแบบทิศทางเดียว และ สองทิศทาง โดยพิจารณาจากทิศทางของหัวลูกศร

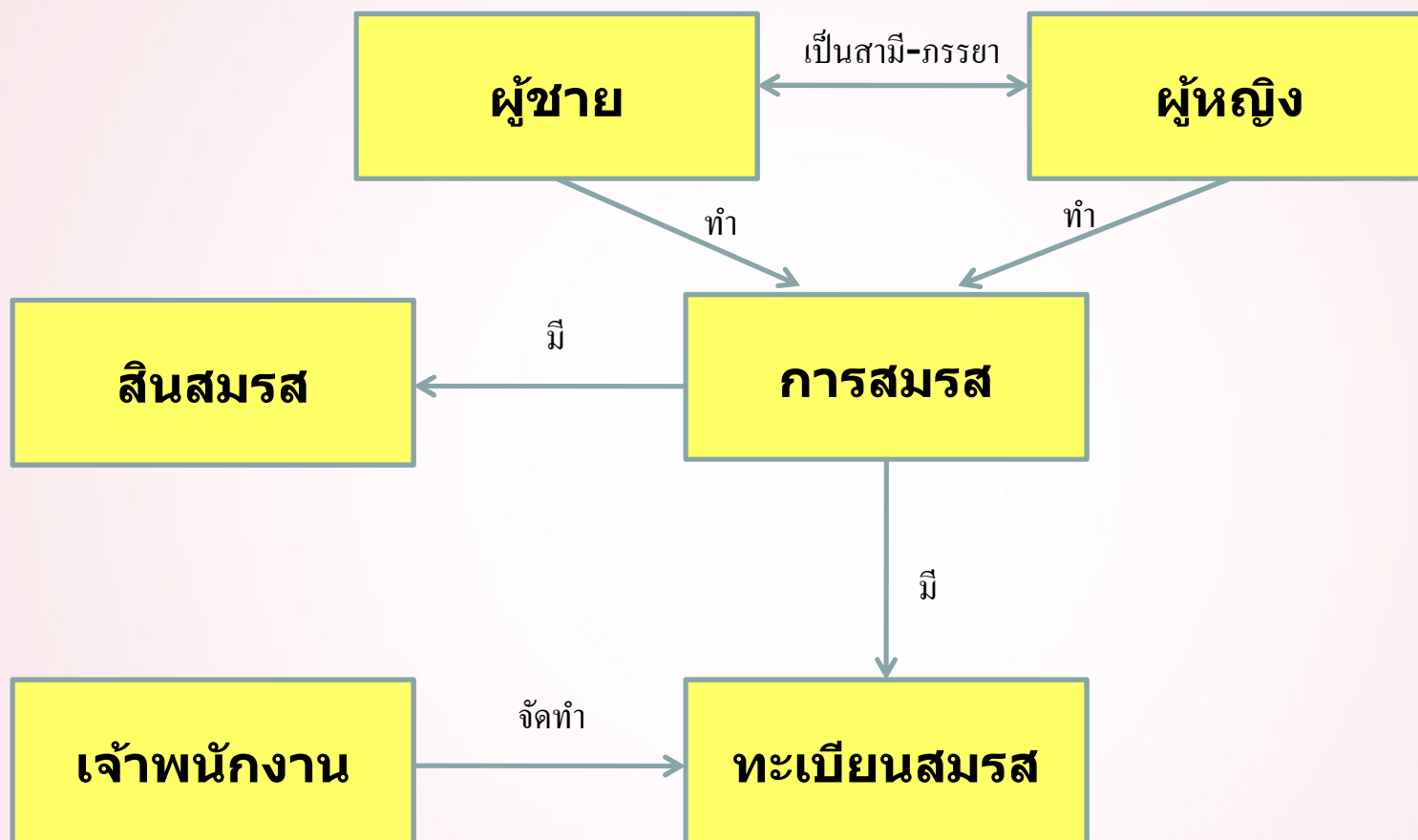
ตัวอย่าง Association



ตัวอย่าง Association



ตัวอย่าง Association



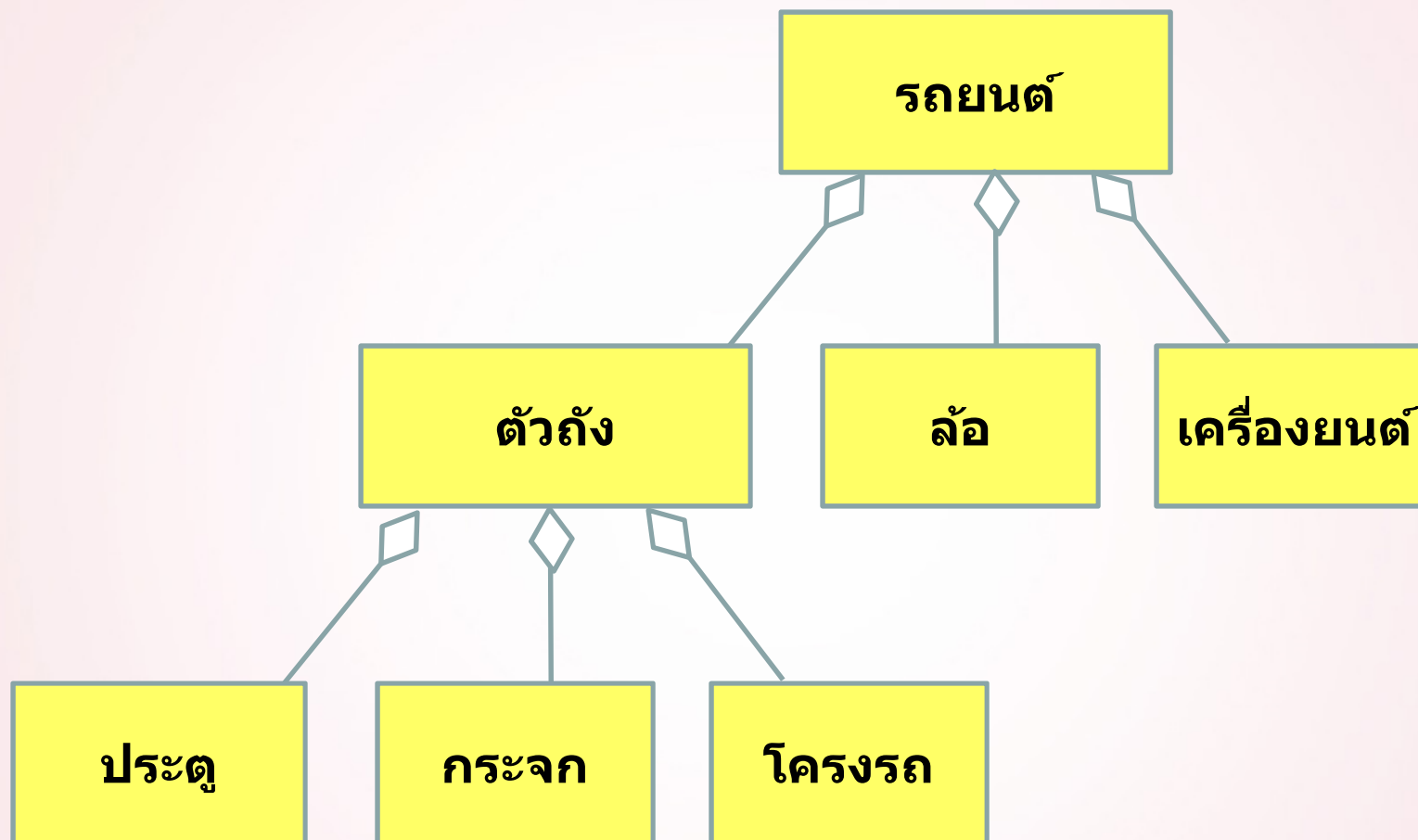
Aggregation

- ❁ เป็นความสัมพันธ์แบบ Association แบบหนึ่ง
- ❁ อาจเรียกว่า whole-part relationship โดยจะมีความสัมพันธ์แบบ “is part of” หรือ “has a”
- ❁ เป็นกระบวนการที่นำเอาคลาสย่อย มาประกอบ (aggregate) กัน เพื่อให้ได้คลาสที่ใหญ่ขึ้น หรือ ซับซ้อนเพิ่มขึ้น
- ❁ คลาสย่อย เรียกว่า Part class ส่วนคลาสที่ประกอบขึ้นจากคลาสย่อย เรียกว่า Whole class
- ❁ การประกอบกันของคลาสย่อย จะต้องทำให้เกิดคลาสใหม่ ซึ่งมี Concept ใหม่เสมอ

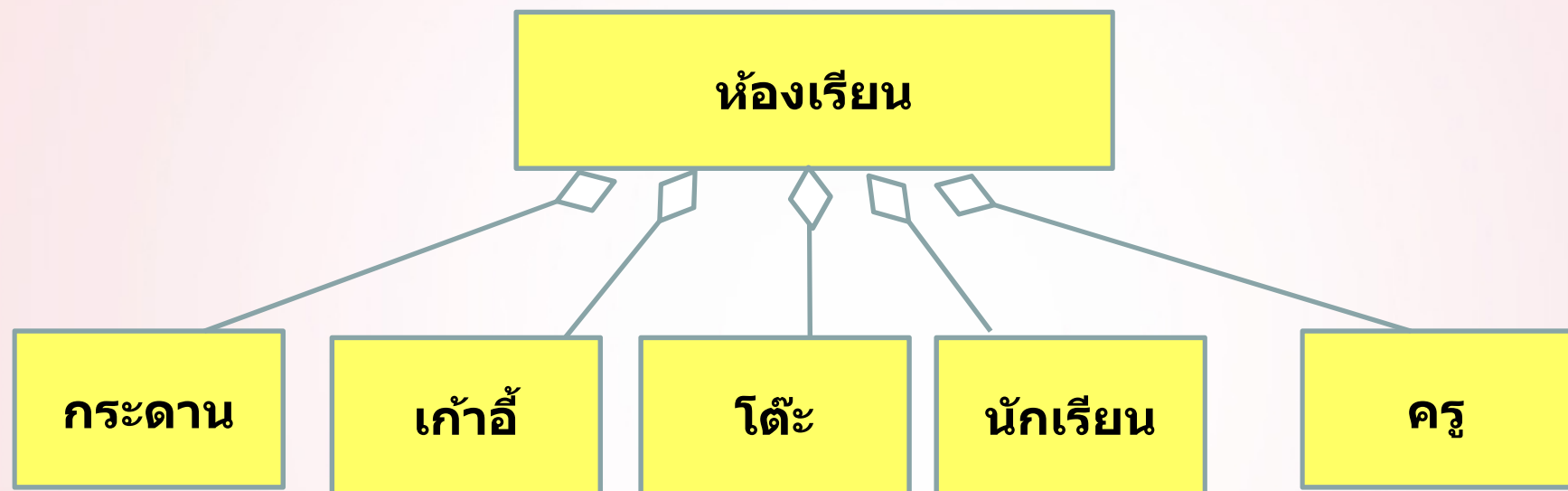
Aggregation

- ❁ เมื่อ Whole class ถูกลบทิ้ง Part class จะยังคงอยู่ได้ โดยไม่ต้องพึ่งพา Whole class
- ❁ คลาส 1 คลาส สามารถนำไปเป็นคลาสย่อยของคลาสอื่น ได้มากกว่า 1 คลาส เรียกว่า **Advanced Aggregation**

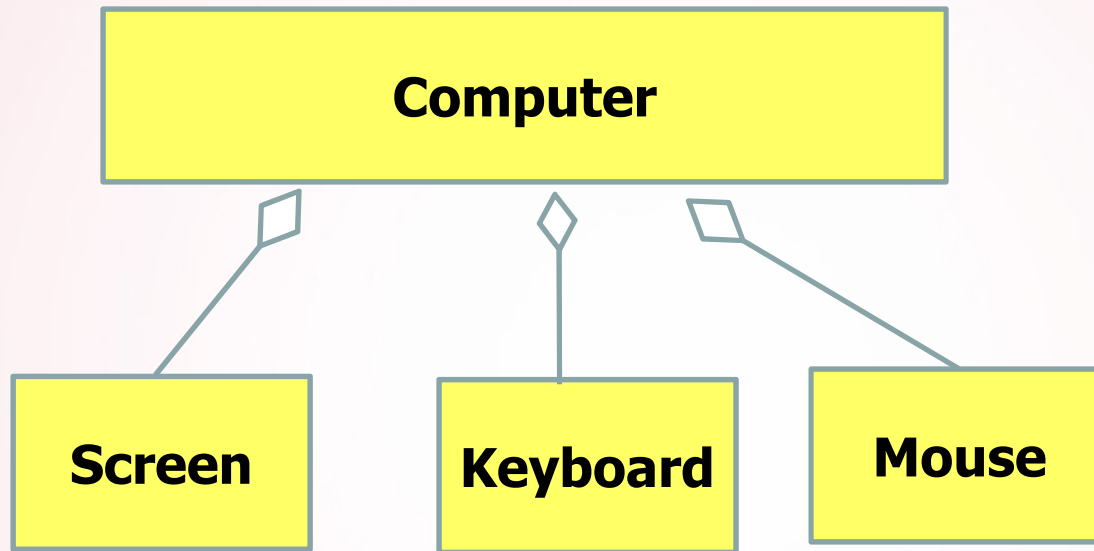
ตัวอย่าง Aggregation : รถยนต์



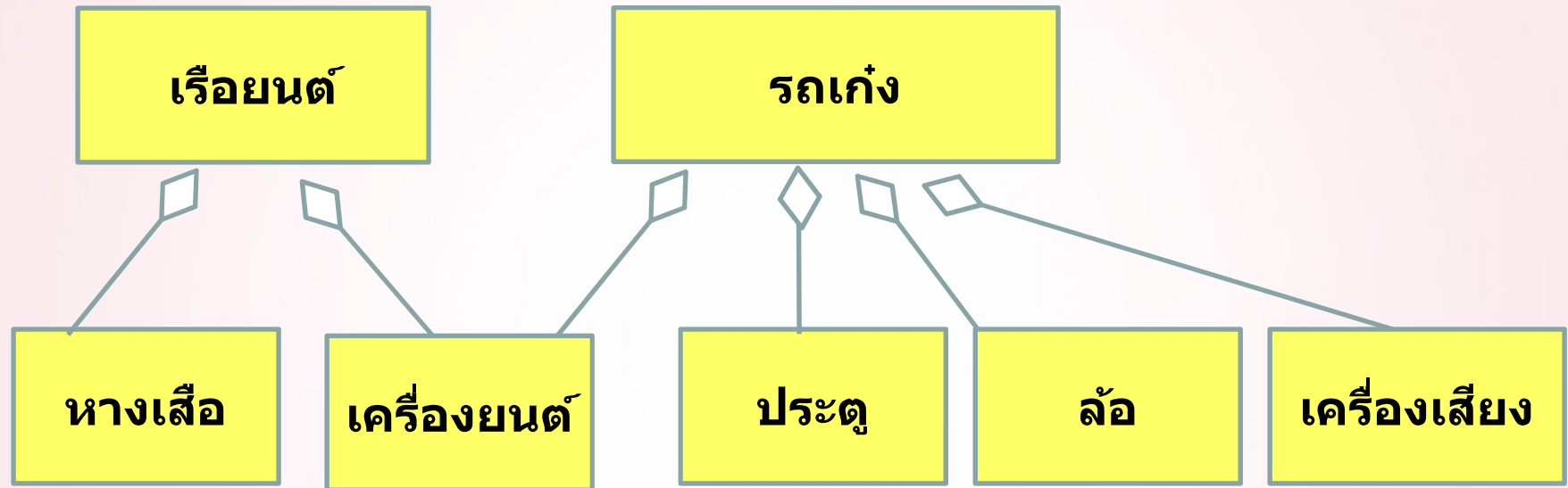
ตัวอย่าง Aggregation : ห้องเรียน



ตัวอย่าง Aggregation



ตัวอย่าง Advanced Aggregation



Cardinality, Required Components and Optional Components

❁ **Cardinality** หมายถึงจำนวนสมาชิกในเซต ซึ่งใช้แสดงจำนวนในความสัมพันธ์

เช่น รถเก๋ง ประกอบด้วยล้อ 4 ล้อ

ประกอบด้วยประตูอย่างน้อย 2 ประตู

❁ **Required Component หรือ Mandatory Components** คือส่วนประกอบที่จำเป็นต้องมี

เช่น รถเก๋ง จำเป็นต้องมี เครื่องยนต์ เสมอ

❁ **Optional Component** คือส่วนประกอบที่ไม่จำเป็นต้องมี เป็นส่วนประกอบที่มีหรือไม่มีก็ได้นั่นเอง

เช่น รถเก๋ง อาจมี หลังคาเปิดประทุน หรือไม่มีก็ได้

Cardinality, Required Components and Optional Components

❁ **Maximum Cardinality** คือจำนวนที่มากที่สุดที่สามารถมีได้

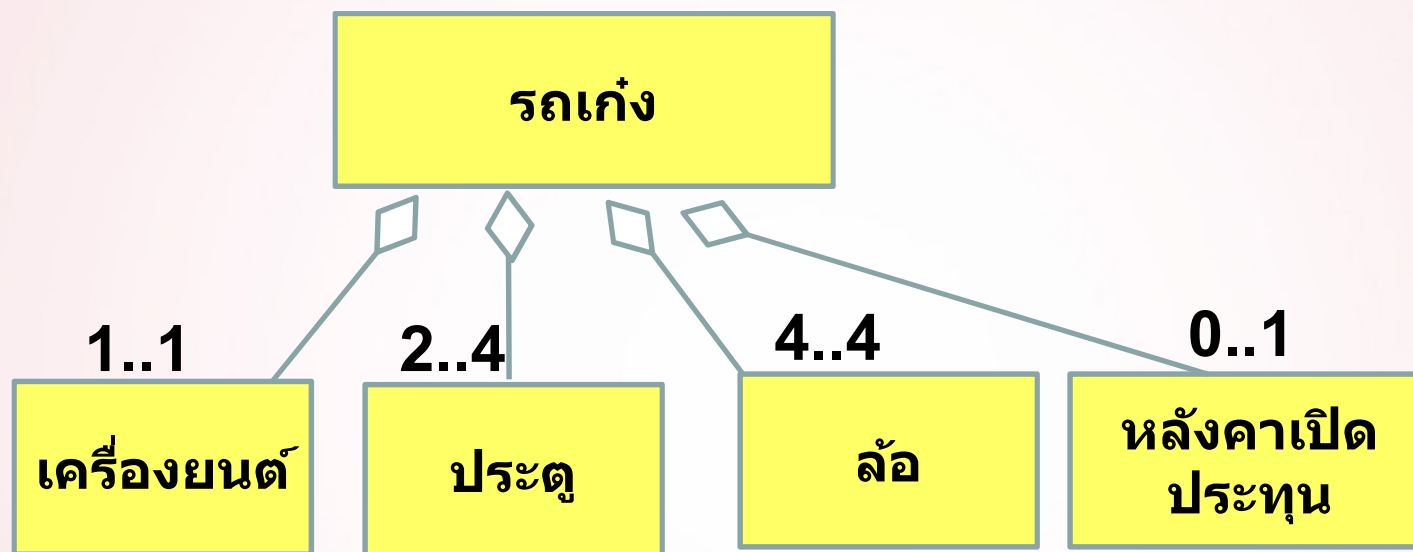
❁ **Minimum Cardinality** คือจำนวนที่น้อยที่สุดที่สามารถมีได้

เช่น รถเก๋ง สามารถมีประตูได้น้อยที่สุด 2 ประตู และมีประตูได้มากที่สุด 4 ประตู

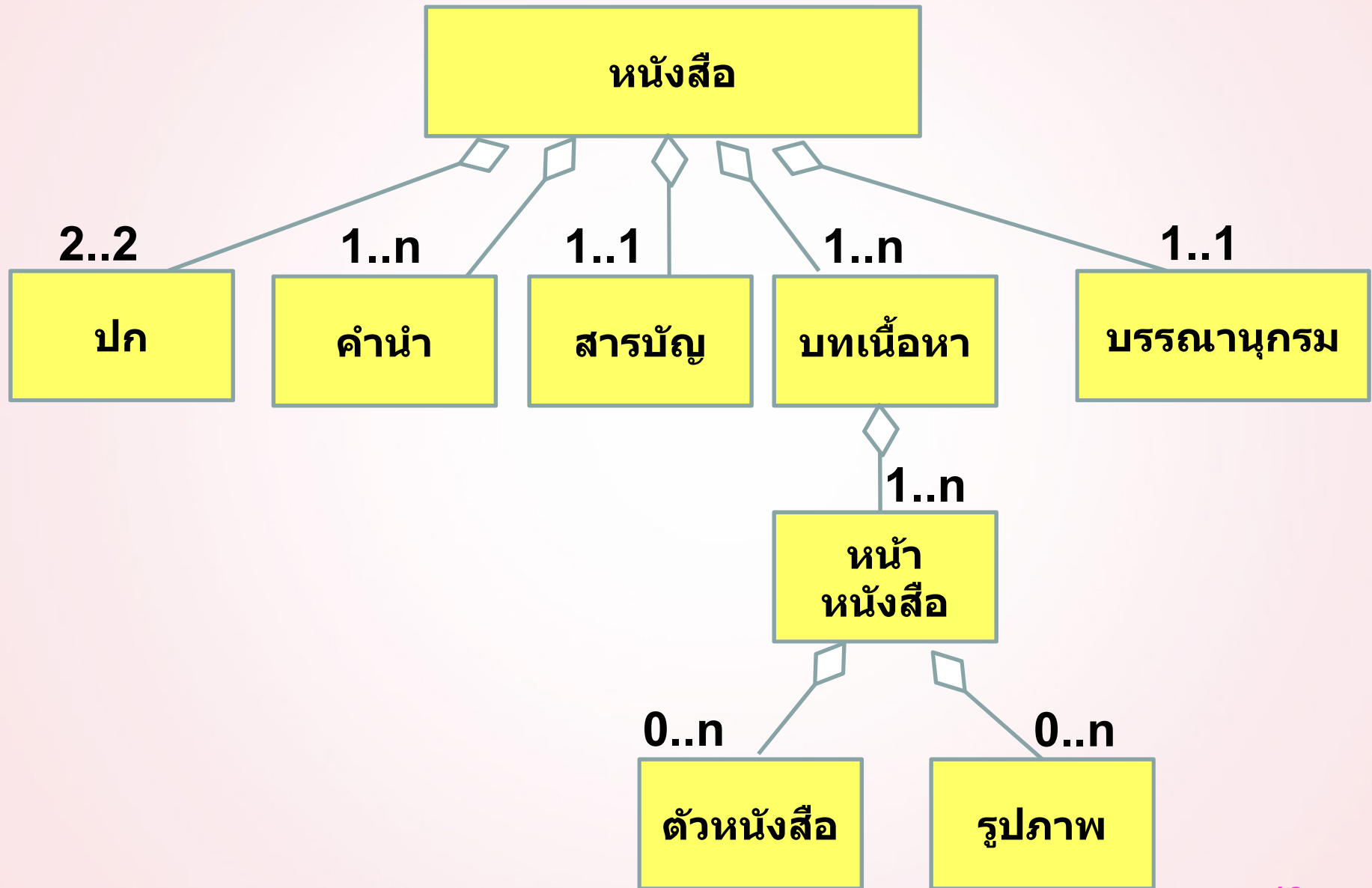
❁ **Required Component** จะมี **Minimum Cardinality** ≥ 1

❁ **Optional Component** จะมี **Minimum Cardinality** ≥ 0

ตัวอย่าง Cardinality : รถเก๋ง



ตัวอย่าง Cardinality : หนังสือ



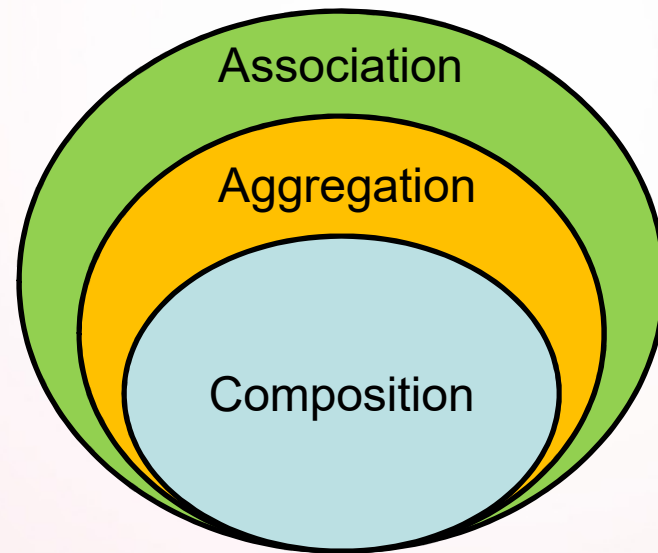
Composition

- ❁ เป็นความสัมพันธ์แบบ **Aggregation** แบบหนึ่ง
- ❁ เป็นความสัมพันธ์ที่เป็น “องค์ประกอบ” ต่อกัน
- ❁ เป็นกระบวนการที่นำเอาคลาสย่อย มาประกอบ (aggregate) กัน เพื่อให้ได้คลาสที่ใหญ่ขึ้น หรือ ซับซ้อนเพิ่มขึ้น เช่นเดียวกับ Aggregation
- ❁ แต่ความแตกต่างคือ ส่วนประกอบย่อยนี้จะไม่ สามารถถูกนำออกจากส่วนประกอบหลักได้ เพราะ ถ้านำส่วนประกอบย่อยออกไปแล้วส่วนประกอบหลัก ก็จะสูญเสียคุณสมบัติของความเป็นคลาสนั้น ๆ จัดเป็นองค์ประกอบที่ไม่อาจขาดได้
- ❁ เมื่อ **Whole class** ถูกลบทิ้ง **Part class** จะต้องถูกลบทิ้งไปด้วย

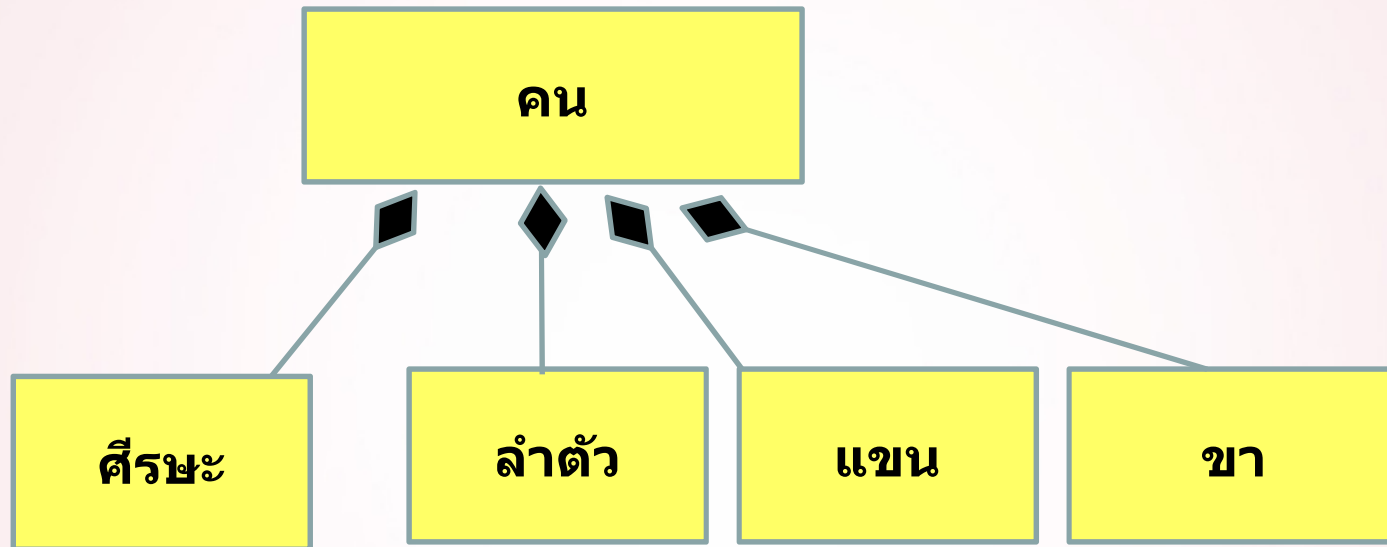
Association, Aggregation, Composition

❁ Aggregation และ Composition เป็นรูปแบบพิเศษของ Association

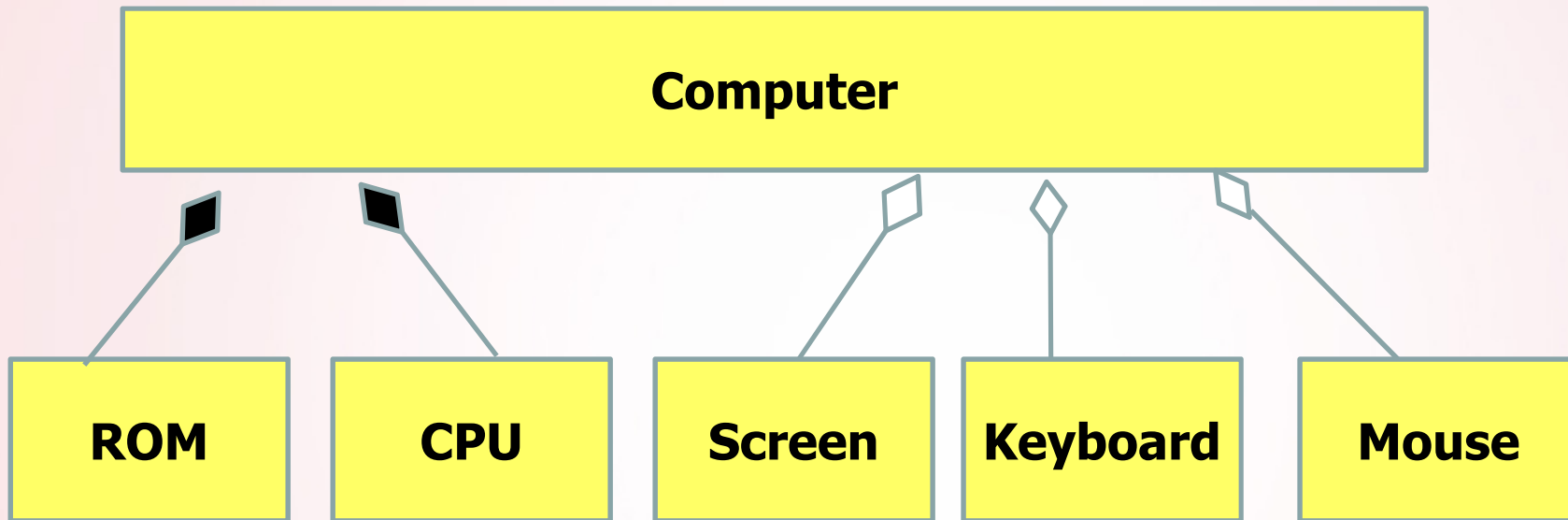
❁ Composition เป็นรูปแบบพิเศษของ Aggregation



ตัวอย่าง Composition



ตัวอย่าง Composition



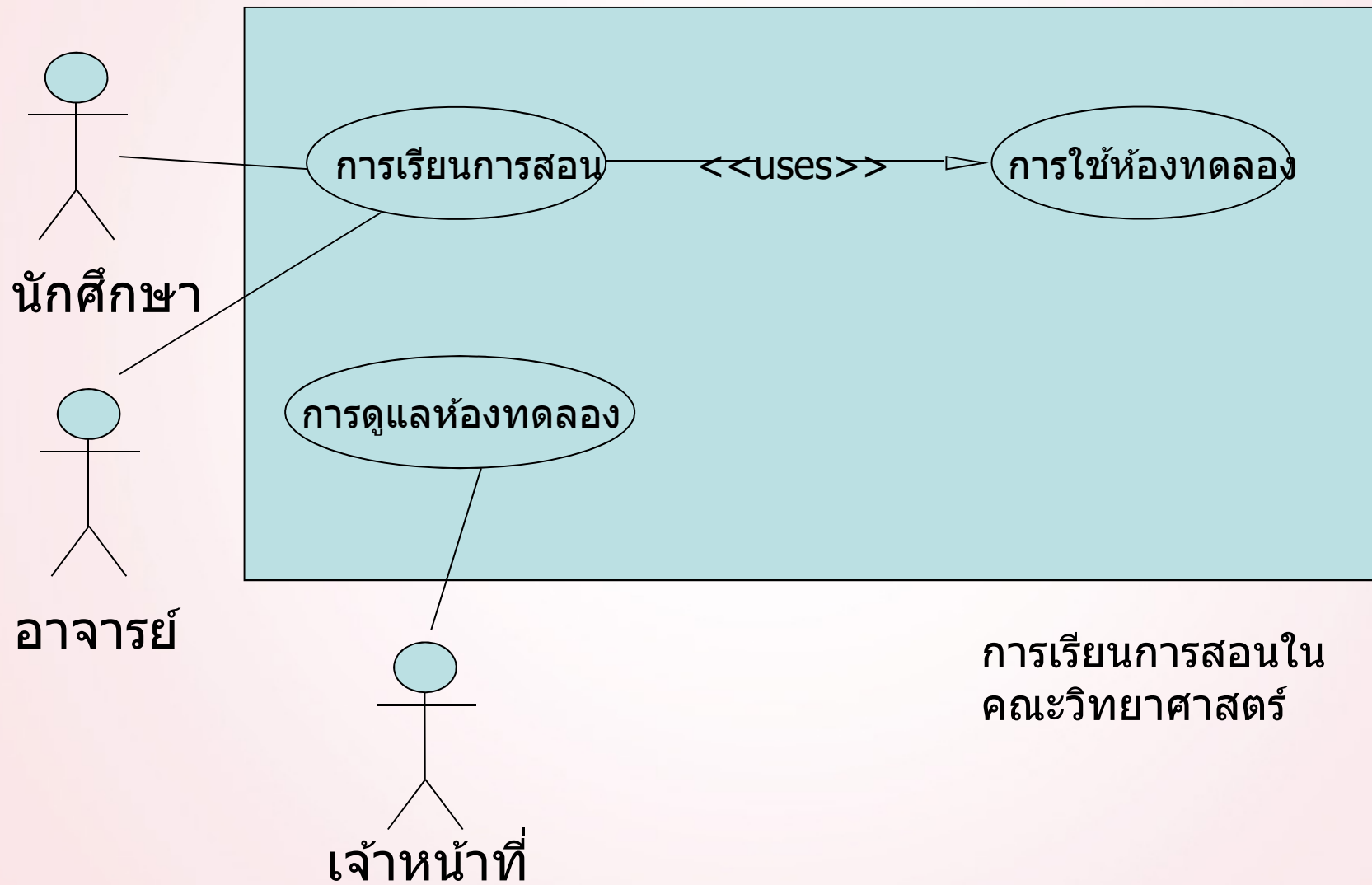
ตัวอย่าง Composition



ตัวอย่าง : จงสร้าง Class Diagram จาก Problem Domain ที่กำหนดให้ต่อไปนี้

- ❁ ในคณะวิทยาศาสตร์ของสถาบันแห่งหนึ่งมีบุคลากรหลายประเภทด้วยกัน ได้แก่ อาจารย์ นักศึกษา และเจ้าหน้าที่
- ❁ อาจารย์แต่ละท่านมีหน้าที่ในการสอนวิชาใดวิชาหนึ่ง หรือมากกว่า 1 วิชาก็ได้
- ❁ นักศึกษามีหน้าที่ในการศึกษาวิชาใดวิชาหนึ่ง หรือมากกว่า 1 วิชาก็ได้ในเวลาเดียวกัน
- ❁ เจ้าหน้าที่ของภาควิชา คือเจ้าหน้าที่ที่ประจำห้องทดลองต่างๆ โดยกำหนดว่าใน 1 ห้องทดลองจะต้องมีเจ้าหน้าที่ประจำ 1 คนเสมอ

Use Case Diagram



Use Case	Object หรือ Class ที่เกี่ยวข้อง
การเรียนรู้การสอน	นักเรียน อาจารย์ ห้องเรียน วิชาเรียน ชั่วโมงเรียน
การใช้ห้องทดลอง	นักเรียน อาจารย์ ห้องทดลอง
การดูแลห้องทดลอง	เจ้าหน้าที่ ห้องทดลอง

Class ที่มีทั้งหมดของระบบได้แก่ **อาจารย์ นักศึกษา
เจ้าหน้าที่ ห้องเรียน วิชาเรียน ชั่วโมงเรียน และห้องทดลอง**

