



คลาสและวัตถุ (Class and Object)

**Matinee Kiewkanya
Computer Science, CMU**

การให้แนวคิดแก้วัตถุ

- ❁ หากพิจารณาสิ่งที่อยู่รอบตัวเราในชีวิตประจำวัน แล้ว เราจะพบ **"วัตถุ"** มากมาย
- ❁ วัตถุที่เป็นสิ่งมีชีวิต เช่น มนุษย์ สัตว์ และ ต้นไม้ เป็นต้น หรือวัตถุที่เป็นสิ่งไม่มีชีวิต เช่น แก้ว ี พัดลม บ้าน โทรศัพท์มือถือ รวมไปถึงวัตถุที่จับต้องไม่ได้ เช่น กฎหมาย รัฐธรรมนูญ เป็นต้น
- ❁ อาจกล่าวได้ว่าคำศัพท์คำใดเป็น **"คำนาม"** แล้ว คำ ๆ นั้นล้วนแล้วแต่เป็นชื่อที่ใช้เรียกวัตถุแทบทั้งสิ้น

คำศัพท์ที่เกี่ยวข้อง

Concept

- ❁ เมื่อกล่าวถึงวัตถุโดยทั่วไปแล้ว จะหมายถึงวัตถุที่อยู่ในชีวิตจริง (Real World)
- ❁ หากเราต้องการจะจำลองวัตถุดังกล่าวให้อยู่ในเครื่องคอมพิวเตอร์ จะต้องมีการให้แนวคิดแก่วัตถุ หรือที่เรียกว่าให้ **Concept** นั้นเอง
- ❁ **Concept** หมายถึง ความคิดรวบยอดที่เรามีให้กับวัตถุนั้น ๆ ภายใต้กรอบ (Domain) ที่กำหนด

คำศัพท์ที่เกี่ยวข้อง

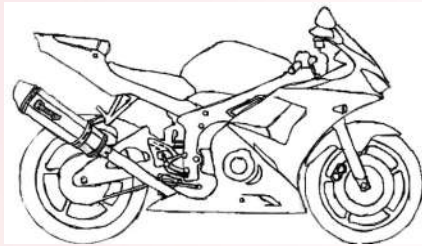
- ❁ เช่น หากโจทย์ปัญหานั้นเกี่ยวข้องกับการลงทะเบียนวิชาต่าง ๆ ของนักศึกษามหาวิทยาลัยเชียงใหม่
- ❁ เราอาจให้แนวคิดแก่วัตถุ “นักศึกษา” ว่าเป็นสิ่งที่มีชื่อ-สกุล มีรหัสนักศึกษา มีบิดา มารดา มีอาจารย์ที่ปรึกษา สามารถลงทะเบียนเรียน สามารถเพิ่ม-ถอน ภาระบวณวิชาที่ลงทะเบียน เป็นต้น
- ❁ และอาจให้แนวคิดแก่วัตถุ “วิชา” ว่าเป็นสิ่งที่มีรหัสภาระบวณ วิชา มีชื่อวิชา มีหน่วยกิต สามารถเปิดสอนหรือปิดสอนในแต่ละเทอม เป็นต้น
- ❁ โดยมีกรอบที่กำหนดคือจะพิจารณาแนวคิดที่มีต่อนักศึกษา และภาระบวณวิชาของมหาวิทยาลัยเชียงใหม่เท่านั้น

คำศัพท์ที่เกี่ยวข้อง

Class

- ❁ ผลจากการให้แนวคิดกับวัตถุในชีวิตจริง จะทำให้เกิดการจำแนกประเภท (Classification) ของวัตถุ ที่เราเรียกว่า **คลาส (Class)**
- ❁ สิ่งที่มีแนวคิดอย่างเดียวกันจะถูกจัดให้อยู่ในคลาสเดียวกัน
- ❁ เช่น ในชีวิตจริงมีการใช้งานวัตถุที่เป็นยานพาหนะประเภทหนึ่ง มียี่ห่อ ยามาฮ่า ฮอนด้า และ คาวาซากิ ซึ่งอาจมีสีสັນและรูปร่างที่แตกต่างกันออกไป แต่วัตถุทั้ง 3 อย่างนี้ล้วนแล้วแต่มีแนวคิดอย่างเดียวกัน คือ มี 2 ล้อ มีเครื่องยนต์ขับเคลื่อนด้วยน้ำมัน สามารถขับขี่ได้

คำศัพท์ที่เกี่ยวข้อง



Class



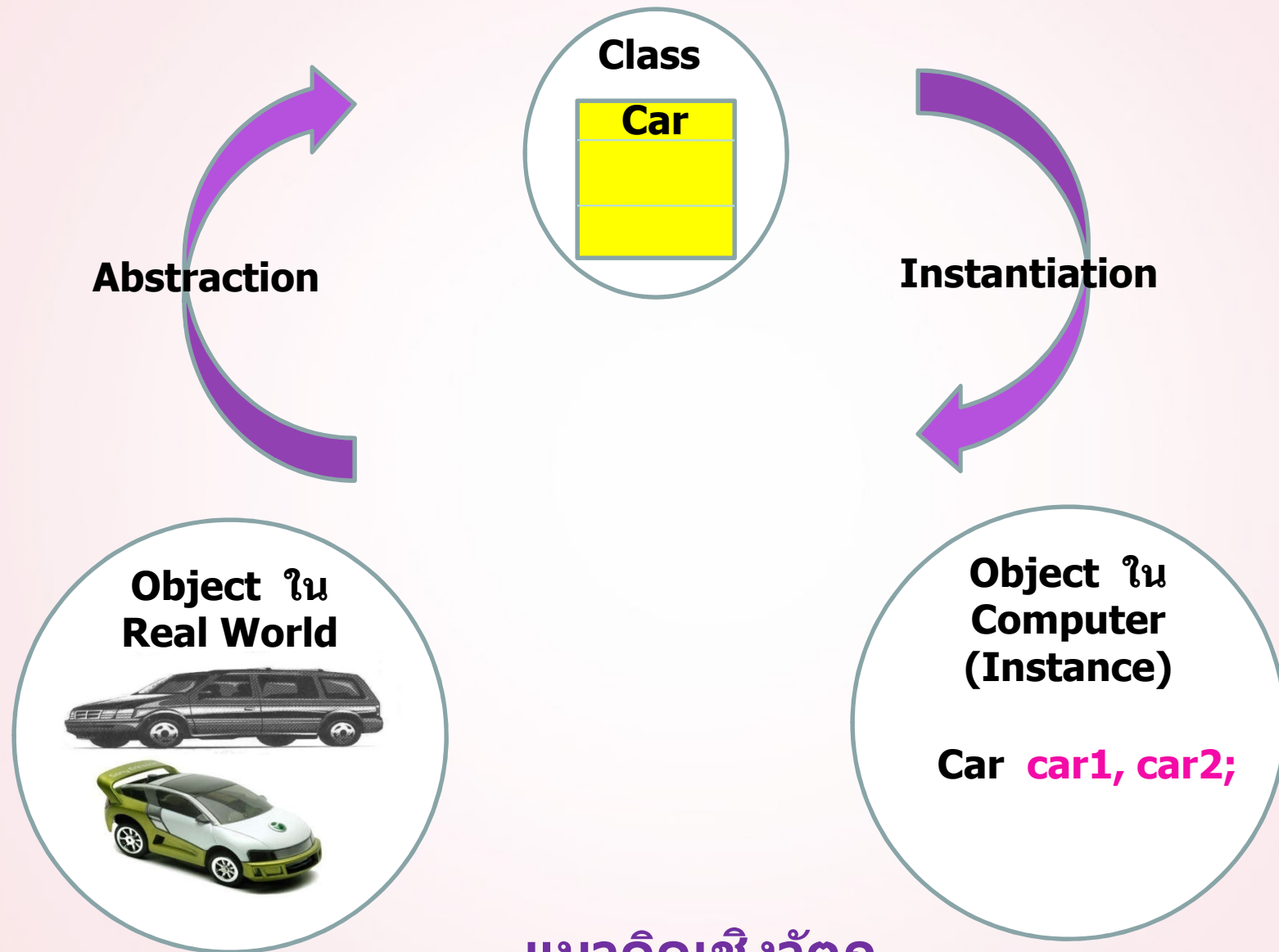
Objects

- ❁ ดังนั้นเราจึงสามารถจัดให้วัตถุทั้งสามอยู่ในคลาสของ "รถจักรยานยนต์" ได้เช่นเดียวกัน
- ❁ กล่าวโดยสรุป คลาส คือ ต้นแบบของวัตถุหรือกลุ่มที่รวมไว้ซึ่งข้อมูลและพฤติกรรม

คำศัพท์ที่เกี่ยวข้อง

Abstraction Instantiation และ Instance

- ✿ **Abstraction** คือ กระบวนการในการให้แนวคิดกับวัตถุในชีวิตจริงจนเกิดเป็นคลาส
- ✿ **Instantiation** คือ กระบวนการสร้างวัตถุในคอมพิวเตอร์จากคลาส
- ✿ **Instance** คือ วัตถุในคอมพิวเตอร์ หรือเรียกอีกอย่างว่าอ็อบเจกต์ (Object) นั้นเอง



แนวคิดเชิงวัตถุ

- ❁ หากเราต้องการเขียนโปรแกรมเชิงวัตถุเพื่อใช้ในการแก้ปัญหาในชีวิตจริง ขั้นตอนแรกจะต้องพิจารณาหาวัตถุในชีวิตจริงที่เกี่ยวข้องกับปัญหาภายใต้กรอบหรือขอบเขตที่เราสนใจ
- ❁ จากนั้นใช้กระบวนการ **Abstraction** เพื่อให้แนวคิดแก่วัตถุในชีวิตจริง จนได้สิ่งที่เรียกว่าคลาส ซึ่งเปรียบเสมือนเป็นต้นแบบของวัตถุในชีวิตจริง
- ❁ แต่คลาสเป็นสิ่งที่ไม่มีตัวตนอยู่จริง ไม่สามารถทำกิจกรรมใด ๆ ได้ เป็นเพียงสิ่งที่อยู่ในความคิดของเราเท่านั้น

❁ ดังนั้นเพื่อให้เราสามารถเขียนโปรแกรมจัดการกับวัตถุนั้นได้ จะต้องทำให้มันมีตัวตนด้วยการสร้างวัตถุในคอมพิวเตอร์หรือที่เรียกว่า **Instance** ขึ้นมาแทนวัตถุในชีวิตจริงด้วยกระบวนการที่เรียกว่า **Instantiation**

❁ ซึ่ง **Instance** นี้เป็นตัวแปรตัวหนึ่งที่จะใช้เขียนโปรแกรมนั่นเอง โดยตัวแปรตัวนี้จะใช้จัดการกับคุณลักษณะและพฤติกรรมของวัตถุที่ **Instance** นั้นเป็นตัวแทนอยู่

อะไรคือคลาส อะไรคือวัตถุ

**“สำนักพิมพ์ KTP จัดพิมพ์หนังสือคัมภีร์จาวาเล่ม 1
และ เล่ม 2”**

คลาส : วัตถุ :

คลาส : วัตถุ :

อะไรคือคลาส อะไรคือวัตถุ

“เกมส์สาวน้อยผจญภัยจะมีตัวการ์ตูนชื่อเจนนี่กับลิลลี่
ที่ในระหว่างการเดินทางต้องคอยหลบระเบิดและการ
ทำร้ายด้วยการปล่อยพลังของศัตรู คือ นกฟีนิกซ์ไฟ
เหี้ยมวร้ายเล็บ และ เมาคลี้จอมโหด”

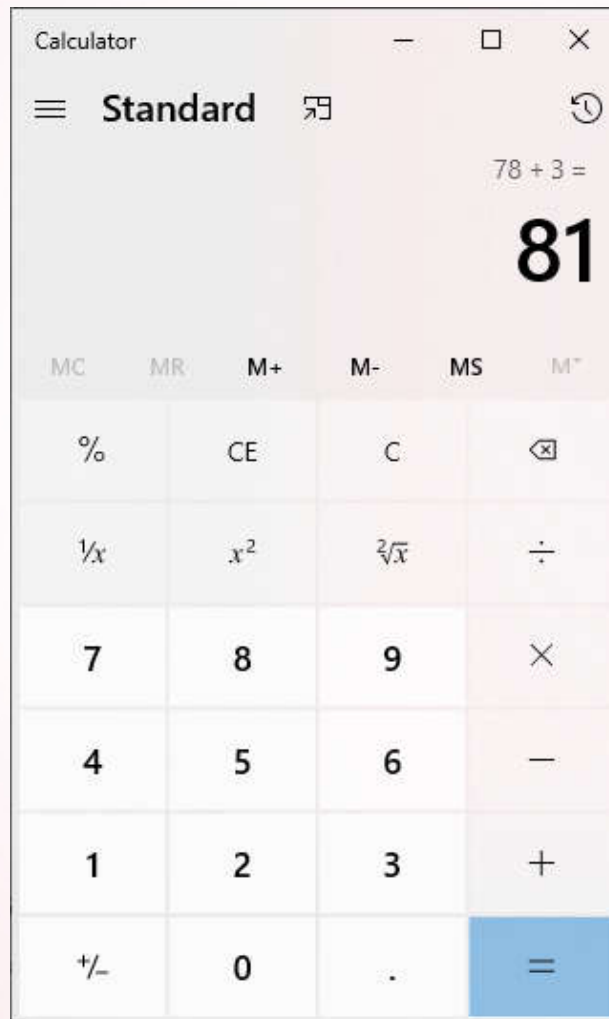
คลาส : วัตถุ :

คลาส : วัตถุ :

คลาส : วัตถุ :

คลาส : วัตถุ :

อะไรคือคลาส อะไรคือวัตถุ



ชนิดข้อมูลแบบคลาส

- ❁ หลักการพื้นฐานของการเขียนโปรแกรมเชิงวัตถุ คือ การสร้างวัตถุขึ้นมาเพื่อทำงานร่วมกันในการแก้ปัญหา โดยปัญหาจะถูกนำมาพิจารณาว่ามีวัตถุใดบ้างที่เกี่ยวข้อง
- ❁ วัตถุในชีวิตจริงจะถูกนำมาให้แนวคิดจนได้เป็นคลาส แล้วจึงทำการสร้างวัตถุในคอมพิวเตอร์ขึ้นมา

ชนิดข้อมูลแบบคลาส

❁ ดังนั้นในการเขียนโปรแกรมเชิงวัตถุจึงต้องมีการนิยามคลาสขึ้นมาก่อน

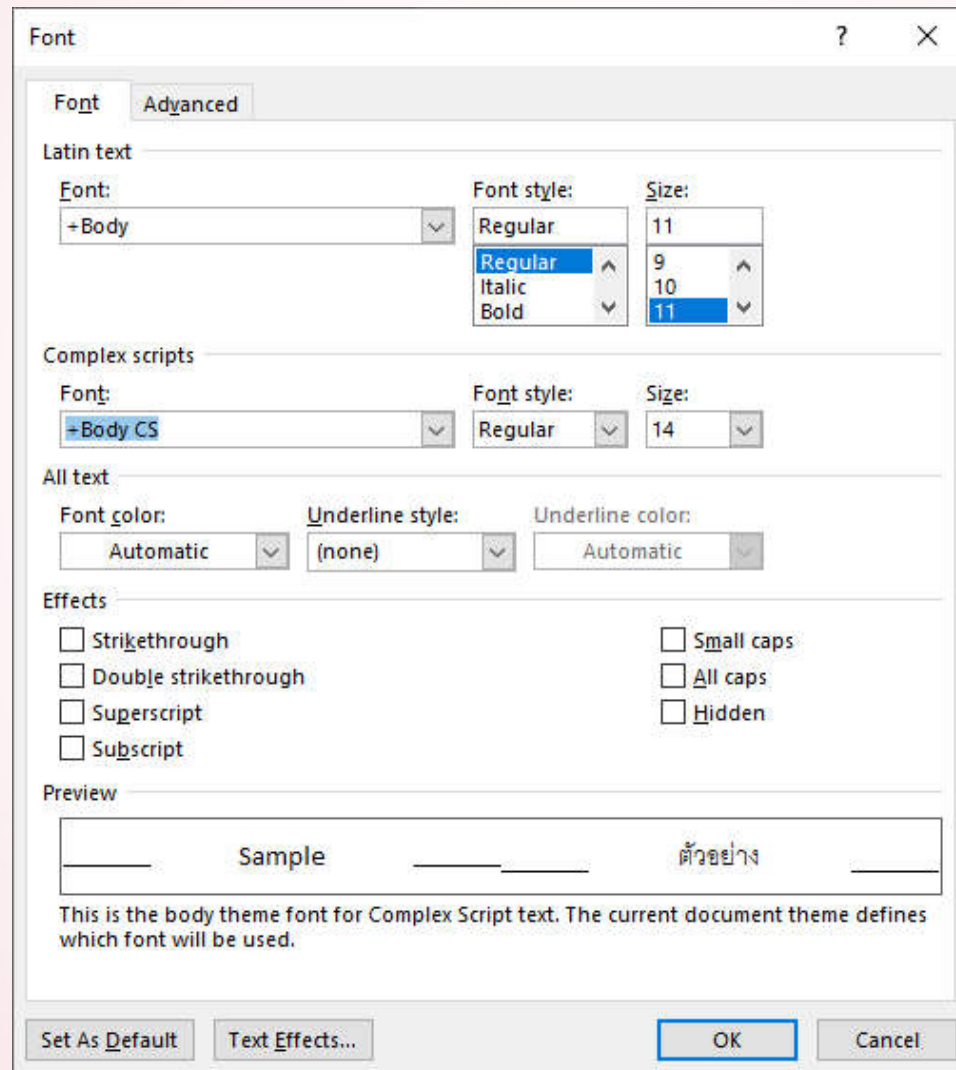
❁ องค์ประกอบของคลาสจะประกอบด้วย 2 ส่วน

1) คุณลักษณะ (Attribute หรือ Data) คือ สิ่งที่บ่งบอกลักษณะและค่าข้อมูลของคลาสนั้น
คุณลักษณะจะบรรยายลักษณะหรือคุณสมบัติของวัตถุต่าง ๆ ได้ ภายใต้กรอบที่เราสนใจ

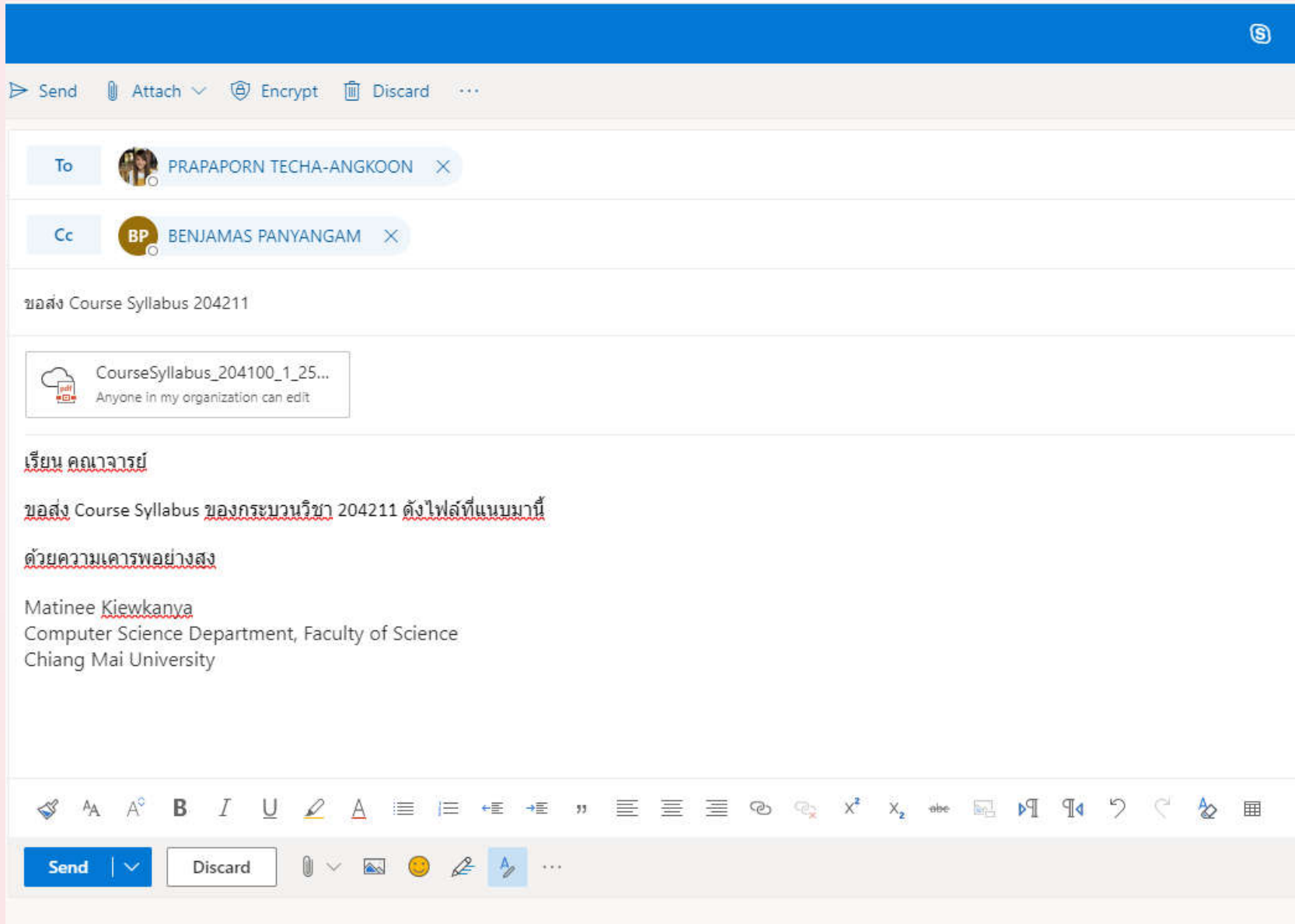
เช่น >> เลขทะเบียน ยี่ห้อ สี เป็นคุณลักษณะของคลาสรถยนต์

>> ชื่อ รหัสนักศึกษา วันเดือนปีเกิด เป็นคุณลักษณะของคลาสนักศึกษา

ตัวอย่างคุณลักษณะของคลาส Font



ตัวอย่างคุณลักษณะของคลาส Email



อะไรคือคุณลักษณะ อะไรคือค่าของคุณลักษณะ?

**"กนกวรรณลงทะเบียนเรียนกระบวนวิชา 204211
(การเขียนโปรแกรมเชิงวัตถุ) เรียนทุกวันจันทร์
และพฤหัสบดี เวลา 12:30-14:30 น."**

พิจารณาคลาสวิชา

คุณลักษณะ : ค่า :

คุณลักษณะ : ค่า :

คุณลักษณะ : ค่า :

คุณลักษณะ : ค่า :

อะไรคือคุณลักษณะ อะไรคือค่าของคุณลักษณะ?

**"ตัวศัตรูที่ชื่อนกฟีนิกซ์ไฟ มีอาวุธคือหางไฟ ที่จะ
คอยปล่อยไฟออกจากหาง โดยระดับความรุนแรง
จะขึ้นอยู่กับค่าพลังในขณะนั้น เช่น ค่าพลัง 100
จะปล่อยไฟได้ 100 ลูก "**

พิจารณาศาสตร์ศัตรู

คุณลักษณะ : ค่า :

คุณลักษณะ : ค่า :

คุณลักษณะ : ค่า :

อะไรคือคลาส อะไรคือวัตถุดิบ ?

อะไรคือคุณลักษณะ อะไรคือค่าของคุณลักษณะ ?



อกเกือบหัก แอบรักคุณสามี
โดย นาวาร้อยกวี (ละครช่อง 3)
นาวาร้อยกวี

Books

★★★★★ 151 ไลค์

~~395.00~~ 335.75 บาท

Add to Cart



เพราะเรา...คู่กัน เล่ม 1-2 (2
เล่มจบ)

JittiRain

Books

★★★★★ 121 ไลค์

~~759.00~~ 607.00 บาท

Add to Cart



ปาฏิหาริย์ร้านข้าวของคุณนาย
ยะ โดย ฮิงาชิโนะ เคโงะ
ฮิงาชิโนะ เคโงะ (Keigo)

Books

★★★★★ 96 ไลค์

~~295.00~~ 250.75 บาท

Add to Cart



สวัสดีครับ คุณภรรยา!

เยว่เซี่ยเตียอิง (Yue Xia Die Ying)

Books

★★★★★ 93 ไลค์

~~315.00~~ 267.75 บาท

Out of Stock

ชนิดข้อมูลแบบคลาส

2) พฤติกรรม (Behavior หรือ Method หรือ Function หรือ Operation) คือ สิ่งที่คลาสสามารถกระทำออกมาได้

เช่น >> การสตาร์ท การเลี้ยว การจอด เป็น
พฤติกรรมของคลาสรถยนต์
>> การลงทะเบียน การเพิ่ม-ถอนกระบวนวิชา
การย้ายคณะ เป็นพฤติกรรมของคลาส
นักศึกษา

ชนิดข้อมูลแบบคลาส

❁ จากประโยคตัวอย่างต่อไปนี้

"ร้านขายสุนัขแห่งหนึ่งมีสุนัขพันธุ์ต่าง ๆ เช่น ชิสุ โกลเด้นรีทรีฟเวอร์ ปอมเมอเรเดียน เป็นต้น ซึ่งจะจัดจำหน่ายด้วยราคาที่แตกต่างกันขึ้นอยู่กับพันธุ์และอายุของสุนัข"

❁ หากเราต้องการเขียนโปรแกรมเพื่อจัดเก็บข้อมูลการขายสุนัขของร้านแห่งนี้ จะต้องทำการกำหนดคลาสที่เกี่ยวข้อง ซึ่งในที่นี้คือคลาส.....

❁ โดยมีคุณลักษณะ คือ.....

❁ และมีพฤติกรรม คือ

ชนิดข้อมูลแบบคลาส

- ❁ วัตถุในโลกของแนวคิดเชิงวัตถุนั้น จะเป็นวัตถุที่ถูกกระทำ (Passive Object)
- ❁ เช่น การเปิดโทรทัศน์ โทรทัศน์ไม่สามารถที่จะเปิดขึ้นมาเองได้ แต่จะต้องมีใครสักคนมาเปิดมัน
- ❁ โดยที่โทรทัศน์จะต้องเตรียมความสามารถในการถูกเปิดเครื่องหรือฟังก์ชันการเปิดเครื่องไว้ เช่นเดียวกับการเปลี่ยนช่อง การปรับเสียง การปรับแสง
- ❁ โดยพฤติกรรมเหล่านี้จะมีการกระทำเมื่อถูกกระตุ้นจากภายนอก

ชนิดข้อมูลแบบคลาส

- ❁ จากตัวอย่างร้านขายสุนัขข้างต้น สุนัขไม่สามารถทำการขายตนเองได้ แต่จะต้องเตรียมฟังก์ชันการขายไว้
- ❁ เมื่อเจ้าของสุนัขได้ขายสุนัขไป จะมีการเรียกใช้ฟังก์ชันนี้เพื่อทำการปรับเปลี่ยนค่าของคุณลักษณะที่เกี่ยวข้อง ซึ่งในที่นี้คือสถานะการขายของสุนัขตัวนั้น ๆ เป็นต้น
- ❁ ดังนั้นพฤติกรรมของคลาสในที่นี้จึงเป็นความสามารถในการทำกิจกรรมที่มีไว้เพื่อให้สิ่งอื่นมากระตุ้นให้เกิดขึ้นได้

นิยามชนิดข้อมูลแบบคลาส

- ❁ **คลาส (Class)** คือ โครงสร้างของข้อมูลชนิดใหม่ มีลักษณะคล้ายข้อมูลชนิด **structure** ในภาษา C
- ❁ **วัตถุ (Object/Instance)** คือ ตัวแปรที่มีชนิดเป็น **class** ในที่นี้จะใช้คำว่า **Instance** ในความหมายเดียวกับคำว่า **Object** โดยจะหมายถึง **Object** ใน **Computer** นั้นเอง
- ❁ การนิยามคลาสจะต้องประกอบด้วย
 - **Member Data** : ตัวแปรที่ใช้เก็บข้อมูลของคลาส โดยอาจเรียกว่า **data** หรือ **attribute** หรือ **property**
 - **Member Method** : เมทอดที่อยู่ภายในคลาส โดยอาจเรียกว่า **function** หรือ **operation**

นิยามชนิดข้อมูลแบบคลาส

❁ โครงสร้างของ Class ประกอบด้วย

- 1. name
- 2. member data
- 3. member methods

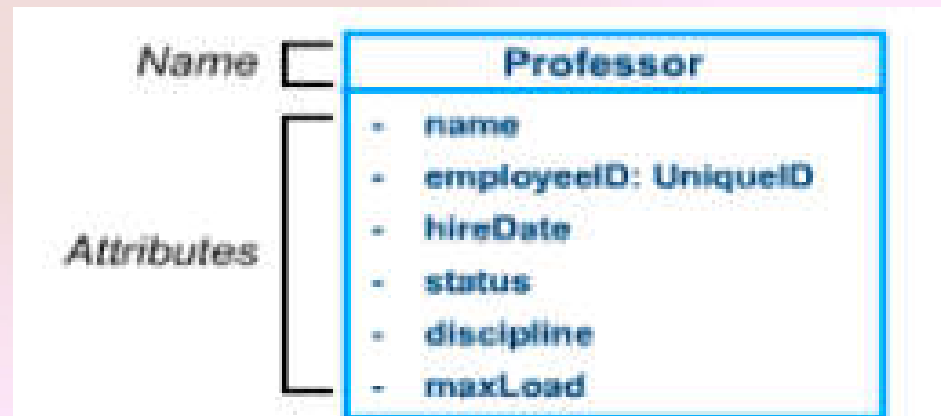
Student	1
- score - grade	2
+setScore() +calGrade() +printGrade()	3

❁ การนิยาม Class เปรียบเสมือนเป็นต้นแบบ (Template) สำหรับการสร้าง Object ของ Class ชนิดนั้น ๆ

❁ Class ทำให้เราสามารถสร้างกลุ่มข้อมูลชนิดใหม่ และการดำเนินการกับกลุ่มข้อมูลนั้น

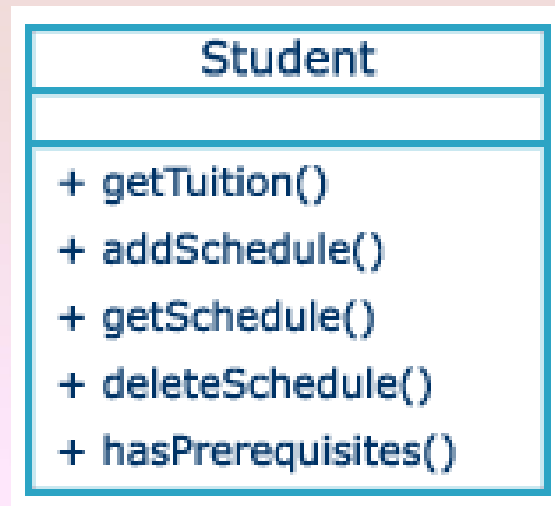
Member Data

- ❁ **Member Data หรือ Attribute** คือชื่อของคุณสมบัติของคลาสหรือเป็นตัวแปรภายในคลาสนั้นเอง โดยจะเป็นสิ่งที่ใช้ในการเก็บค่าหรือสถานะของ **Object** (instance of class)
- ❁ **Member Data จะมีชนิด(type)** ตามประเภทข้อมูลที่จัดเก็บ ซึ่งอาจจะเป็นชนิดพื้นฐาน เช่น **integer, float, char** หรือ อาจจะมีชนิดเป็นคลาสก็ได้



Member Method

- ❁ **Member Method** คือบริการที่คลาสนั้นๆมีไว้ให้ **Object** ของคลาสนั้นเรียกใช้งานได้
- ❁ **Member Method** จะเป็นตัวกำหนดว่าคลาสนั้นสามารถทำอะไรหรือให้บริการอะไรได้บ้าง



รูปแบบการนิยามชนิดข้อมูลแบบคลาส

```
[modifier] class ClassName {  
    [ [modifier] member_data;  
    "[modifier] member_method;  
    ]"  
}
```

โดยที่

- ❁ **modifier** คือ คำสำคัญ ที่ใช้ระบุระดับการเข้าถึงได้แก่ **private**, **package**, **protected**, **public** หรือคำอธิบายคุณสมบัติอื่นๆ เช่น **abstract** และ **final**
- ❁ **class** คือ คำสำคัญ ที่ระบุว่าเป็นการประกาศคลาส
- ❁ **ClassName** คือชื่อคลาส นิยมตั้งชื่อขึ้นต้นด้วยตัวพิมพ์ใหญ่
- ❁ **member_data** คือการนิยามข้อมูลที่เป็นสมาชิกของคลาส
- ❁ **member_method** คือการนิยามเมทอดที่เป็นสมาชิกของคลาส

ตัวดัดแปร (Modifier)

❁ เป็นคำสำคัญที่ใช้ระบุการเข้าถึงสมาชิกของคลาส

	ใช้ได้ ทั้งหมด	package เดียวกัน	ต่าง package กัน	ต่าง package กัน แต่เป็นคลาส แม่ คลาสลูกกัน	คลาส เดียว กัน
public	✓	✓	✓	✓	✓
protected	✗	✓	✗	✓	✓
package	✗	✓	✗	✗	✓
private	✗	✗	✗	✗	✓

private, package, protected, public

(เรียงลำดับจากความเข้มงวดมากถึงน้อย (อิสระที่สุด))

รูปแบบการนิยามเมทอดสมาชิก

```
[modifier] return_type methodName ([parameters]) {  
    [body]  
}
```

โดยที่

- ✿ **modifier** คือ คำสำคัญที่ใช้ระบุระดับการเข้าถึง
- ✿ **return_type** คือ ชนิดข้อมูลที่เมทอดจะคืนค่ากลับ
- ✿ **methodName** คือ ชื่อเมทอดสมาชิก
- ✿ **parameter** คือ ตัวแปรที่กำหนดขึ้นเพื่อรับข้อมูลเข้าจากผู้เรียกใช้
- ✿ **body** คือ ส่วนกำหนดการทำงานของเมทอด ที่ประกอบด้วย
ตัวแปรเฉพาะที่ (Local Variable) และคำสั่งกำหนดการทำงาน

ตัวอย่างที่ 1 : การนิยามคลาส Grade

```
public class Grade{  
    private int  score;  
    private char grade;  
  
    public void setScore(int s){  
        score = s;  
    }  
    public void calGrade(){  
        if (score > 50)  
            grade = 'S';  
        else  
            grade = 'U';  
    }  
    public char getGrade(){  
        return(grade);  
    }  
}
```

ตัวอย่างที่ 2 : การนิยามคลาส Student

```
public class Student{  
    private string id;  
    private string name;  
    private double gpa;  
  
    public void setID(string ID){  
        id = ID;  
    }  
    public void setName(string n){  
        name = n;  
    }  
    public void setGPA(double GPA){  
        gpa = GPA;  
    }  
    public void showDetails(){  
        System.out.println("ID : "+id);  
        System.out.println("Name : "+name);  
        System.out.println("GPA : "+gpa);  
    }  
}
```

การประกาศ object

```
[modifier] ClassName objectName;
```

โดยที่

- ❁ **modifier** คือ access modifier เช่น public หรือคำอธิบายคุณสมบัติอื่นๆ ของคลาส เช่น abstract และ final
- ❁ **ClassName** คือชื่อของ class
- ❁ **objectName** คือชื่อของ object ที่เป็นชื่อ identifier ใดๆ ที่สอดคล้องกับกฎการตั้งชื่อ

ตัวอย่าง

Student s1;

การสร้าง object

```
objectName = new ClassName(parameters);
```

โดยที่

- ❁ **objectName** คือ ชื่อของ object
- ❁ **new** คือ keyword ใช้ในการสร้าง object
- ❁ **ClassName** คือ ชื่อของ class
- ❁ **parameters** คือ ข้อมูลที่ส่งไปเพื่อสร้าง object โดยอาจมีมากกว่าหนึ่งตัว หรือไม่มีเลยก็ได้ขึ้นอยู่กับข้อกำหนด constructor method

ตัวอย่าง

```
s1 = new Student();
```

การรวมคำสั่งสำหรับประกาศและสร้าง object

```
[modifier] ClassName objectName =  
new ClassName([parameters]);
```

ตัวอย่าง

```
Student s1 = new Student();
```


รูปแบบการอ้างถึงข้อมูลสมาชิก
objectName.attributeName

รูปแบบการเรียกใช้เมทอดสมาชิก
objectName.methodName ([parameters])

โดยที่

- ❁ **objectName** คือ ชื่อวัตถุของคลาส
- ❁ **attributeName** คือ ชื่อ attribute หรือ data
- ❁ **methodName** คือ ชื่อเมทอดที่ต้องการเรียกใช้
- ❁ **arguments** คือ ข้อมูลที่ส่งให้แก่เมทอด ซึ่งอาจเป็นตัวแปร

ค่าคงที่ หรือนิพจน์

ตัวอย่าง

การอ้างถึงข้อมูล เช่น

s1.score = 30;

การเรียกใช้เมทอด เช่น

s1.setName("Nid"); ³⁷

การสร้าง object

- ❁ ในหนึ่งคลาสสามารถสร้าง object ใช้งานได้หลายครั้ง
- ❁ แต่ละครั้งที่มีการสร้าง object ด้วย keyword new จะมีการประกาศขอใช้พื้นที่หน่วยความจำใหม่
- ❁ หาก object ใด มีการ new แล้วไม่ถูกใช้งาน ก็จะทำให้เกิดการสะสมขยะขึ้น
- ❁ จาวาจะมีการจัดการคืนพื้นที่หน่วยความจำให้กับระบบ

ตัวอย่างโปรแกรมที่ 1

โปรแกรมเพื่อรับข้อมูลคะแนนวิชาสัมมนาของ นักศึกษา
1 คน แล้วหาว่านักศึกษานั้นได้เกรดอะไร โดยหาก
คะแนนต่ำกว่า 50 ให้ได้เกรด U มิฉะนั้นแล้ว ให้ได้
เกรด S

วิเคราะห์โจทย์

- ❁ วัตถุในชีวิตจริงที่เกี่ยวข้องกับโจทย์คือ **นักศึกษา**
- ❁ เราต้องนำวัตถุในชีวิตจริงมาใส่ในคอมพิวเตอร์ด้วยการนิยามคลาสนักศึกษาขึ้นมาก่อน แล้วจึงสร้างวัตถุในคอมพิวเตอร์ขึ้นมา 1 วัตถุ สำหรับใช้แทนนักศึกษา 1 คน เรียกว่า 1 Object หรือเรียกว่า 1 Instance
- ❁ นักศึกษาจะมี**คุณลักษณะหรือข้อมูล**ที่เกี่ยวข้องกับโจทย์ คือ คะแนน และเกรด
- ❁ นักศึกษาจะมี**พฤติกรรม**ที่เกี่ยวข้องกับโจทย์ คือ การรับคะแนน การตัดเกรด และการพิมพ์เกรด

วิเคราะห์โจทย์

- ❁ ดังนั้นการนิยามคลาสนักศึกษา (Class Student) จะประกอบด้วย
 - ข้อมูลสมาชิก คือ คะแนน (score) และ เกรด (grade)
 - เมทอดสมาชิก คือ การรับคะแนน (setScore) การตัดเกรด (calGrade) และการพิมพ์เกรด (printGrade)

โปรแกรม

```
class Student {           //class definition
    private int score;
    private char grade;
    public void setScore() { //method definition
        Scanner inVar = new Scanner(System.in);
        System.out.print("Enter score : ");
        score = inVar.nextInt();
    }
    public void calGrade() { //method definition
        if (score < 50)
            grade = 'U' ;
        else
            grade = 'S' ;
    }
    public void printGrade() {
        System.out.println("Your score is " + score);
        System.out.println(" Your grade is "+grade);
    }
} //end class Student
```

```
public class JavaApp1 {  
    public static void main(String[] args) {  
        Student stu = new Student(); // Instantiation  
  
        stu.setScore();           // Message Passing  
        stu.calGrade();           // Message Passing  
        stu.printGrade();         // Message Passing  
    }  
} //end class JavaApp1
```

ตัวอย่างผลลัพธ์

Object stu

score

grade

run ครั้งที่ 1

Enter score : 80 ↵

Your score is 80

Your grade is S

run ครั้งที่ 2

Enter score : 45 ↵

Your score is 45

Your grade is U

ตัวอย่างโปรแกรมที่ 2

โปรแกรมเพื่อรับข้อมูลคะแนนวิชาสัมมนาของ นักศึกษา 2 คน แล้วหาว่านักศึกษาแต่ละคนได้เกรดอะไร โดยหากคะแนนต่ำกว่า 50 ให้ได้เกรด U มิฉะนั้นแล้ว ให้ได้เกรด S

โจทย์ในตัวอย่างนี้ เป็นโจทย์เดียวกันกับตัวอย่างโปรแกรม ที่ 1 แต่จะแตกต่างกันที่โจทย์ในตัวอย่างโปรแกรมที่ **1** เป็นการกระทำกับนักศึกษา 1 คน แต่โจทย์ข้อนี้เป็นการกระทำกับนักศึกษา 2 คน

วิเคราะห์โจทย์

- ❁ ในการนิยามคลาสใด ๆ จะต้องพิจารณาจากคุณลักษณะ และพฤติกรรมของวัตถุในชีวิตจริงเพียง 1 วัตถุเท่านั้น ส่วนการสร้างวัตถุในคอมพิวเตอรืขึ้นมาจากคลาส จะขึ้นอยู่กับว่าในโจทย์นั้นมีวัตถุจำนวนเท่าไร
- ❁ วัตถุในชีวิตจริงที่เกี่ยวข้องกับโจทย์คือ **นักศึกษา**
- ❁ เราต้องนำวัตถุในชีวิตจริงมาใส่ในคอมพิวเตอรืด้วยการนิยามคลาสนักศึกษาขึ้นมาก่อน แล้วจึงสร้างวัตถุในคอมพิวเตอรืขึ้นมา 2 วัตถุ สำหรับใช้แทนนักศึกษา 2 คน เรียกว่า 2 Object หรือเรียกว่า 2 Instance
- ❁ นักศึกษาแต่ละคนจะมีคุณลักษณะหรือข้อมูลที่เกี่ยวข้องกับโจทย์คือ คะแนน และเกรด
- ❁ นักศึกษาแต่ละคนจะมีพฤติกรรมที่เกี่ยวข้องกับโจทย์ คือ การรับคะแนน การตัดเกรด และการพิมพ์เกรด

วิเคราะห์โจทย์

- ❁ การนิยามคลาสนักศึกษา จะพิจารณาจากคุณลักษณะและพฤติกรรมของนักศึกษาหนึ่งคนใด ๆ เท่านั้น โดยคลาสนักศึกษาจะเป็นต้นแบบสำหรับนักศึกษาทุกคน ไม่ว่านักศึกษาคนใดก็ตามจะต้องมีคุณลักษณะและพฤติกรรมเช่นเดียวกันกับที่ได้นิยามคลาสไว้ ส่วนการสร้างวัตถุจะสร้างตามจำนวนที่โจทย์กำหนด เช่นตัวอย่างนี้จะมีการสร้างวัตถุขึ้นมา 2 วัตถุ สำหรับใช้แทนนักศึกษา 2 คน
- ❁ ดังนั้นการนิยามคลาสนักศึกษา (Class Student) จะประกอบด้วย
 - ข้อมูลสมาชิก คือ คะแนน (score) และ เกรด (grade)
 - เมทอดสมาชิก คือ การรับคะแนน (setScore) การตัดเกรด (calGrade) และการพิมพ์เกรด (print Grade)

```
import java.util.Scanner;
class Student {           //class definition
    private int score;
    private char grade;
    public void setScore(int i) {           //method definition
        Scanner inVar = new Scanner(System.in);
        System.out.print("Enter score of student no. "+ i + ":");
        score = inVar.nextInt();
    }
    public void calGrade() { //method definition
        if (score < 50)
            grade = 'U' ;
        else
            grade = 'S' ;
    }
    public void printGrade(int i) {
        System.out.println("Score of student no. "+ i + " is " + score);
        System.out.println("Grade of student no. "+ i + " is " + grade + "\n");
    }
} //end class Student
```

```
public class JavaApp2 {  
    public static void main(String[] args) {  
        Student  stu1, stu2;  
  
        stu1 = new Student();  
        stu1.setScore(1);    // Message Passing  
        stu1.calGrade();    // Message Passing  
        stu1.printGrade(1); // Message Passing  
  
        stu2 = new Student();  
        stu2.setScore(2);    // Message Passing  
        stu2.calGrade();    // Message Passing  
        stu2.printGrade(2); // Message Passing  
  
    }  
} //end class JavaApp2
```

ตัวอย่างผลลัพธ์

Object stu1

score	45
grade	U

Object stu2

score	98
grade	S

Enter score of student no. 1: 45
Score of student no. 1 is 45
Grade of student no. 1 is U

Enter score of student no. 2: 98
Score of student no. 2 is 98
Grade of student no. 2 is S

ตัวอย่างการออกแบบคลาสที่ถูกต้องและผิดพลาด

การนิยาม Class และ
สร้าง Object ที่ถูกต้อง

```
class Student{  
    private int score;  
    private char grade;  
  
    public void setScore( );  
    public void calGrade( );  
    public void printGrade( );  
};
```

```
Student Stu1, Stu2 ;
```



สร้าง 2 objects โดยให้ แต่ละ
object เก็บข้อมูลของนักเรียน
แต่ละคน

การนิยาม Class และ
สร้าง Object ที่ผิด

```
class Student{  
    private int score1, score2;  
    private char grade1, grade2;  
  
    public void setScore( );  
    public void calGrade( );  
    public void printGrade( );  
};
```

```
Student Stu;
```

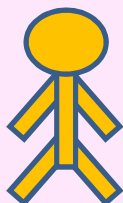
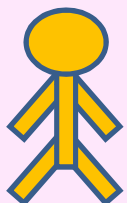


สร้าง object เดียว โดยให้เก็บข้อมูล
ของนักเรียนทั้ง 2 คน

ให้ลองเทียบเคียงกับการประมวลผลกับคน 2 คนดังนี้

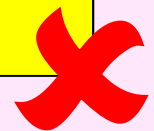
```
class Human{  
    private หัว 1 หัว  
    private แขน 2 แขน  
    private ขา 2 ขา  
  
    public .....  
  
};
```

Human Hum1, Hum2 ;



```
class Human {  
    private หัว 2 หัว  
    private แขน 4 แขน  
    private ขา 4 ขา  
  
    public .....  
  
};
```

Human Hum;



ตัวอย่างโปรแกรมที่ 3

โปรแกรมเพื่อรับข้อมูลคะแนนวิชาสัมมนาของ นักศึกษา N คน แล้วหาว่านักศึกษาแต่ละคนนั้นได้เกรดอะไร โดยหากคะแนนต่ำกว่า 50 ให้ได้เกรด U มิฉะนั้นแล้ว ให้ได้เกรด S

โจทย์ในตัวอย่างนี้ คล้ายกับตัวอย่างโปรแกรมที่ 1 และ 2 แต่จะแตกต่างกันที่โจทย์ในตัวอย่างโปรแกรมที่ **1** และ 2 เป็นการกระทำกับนักศึกษา 1 คน และ 2 คน ตามลำดับ แต่โจทย์ข้อนี้เป็นการกระทำกับนักศึกษา **N** คน

วิเคราะห์โจทย์

- ❁ ในการนิยามคลาสใด ๆ จะต้องพิจารณาจากคุณลักษณะ และพฤติกรรมของวัตถุในชีวิตจริงเพียง 1 วัตถุเท่านั้น ส่วนการสร้างวัตถุในคอมพิวเตอรืขึ้นมาจากคลาส จะขึ้นอยู่กับว่าในโจทย์นั้นมีวัตถุจำนวนเท่าไร
- ❁ วัตถุในชีวิตจริงที่เกี่ยวข้องกับโจทย์คือ **นักศึกษา**
- ❁ เราต้องนำวัตถุในชีวิตจริงมาใส่ในคอมพิวเตอรืด้วยการนิยามคลาสนักศึกษาขึ้นมาก่อน แล้วจึงสร้างวัตถุในคอมพิวเตอรืขึ้นมา N วัตถุ สำหรับใช้แทนนักศึกษา N คน เรียกว่า N Object หรือเรียกว่า N Instance
- ❁ นักศึกษาแต่ละคนจะมีคุณลักษณะหรือข้อมูลที่เกี่ยวข้องกับโจทย์คือ คะแนน และเกรด
- ❁ นักศึกษาแต่ละคนจะมีพฤติกรรมที่เกี่ยวข้องกับโจทย์ คือ การรับคะแนน การตัดเกรด และการพิมพ์เกรด

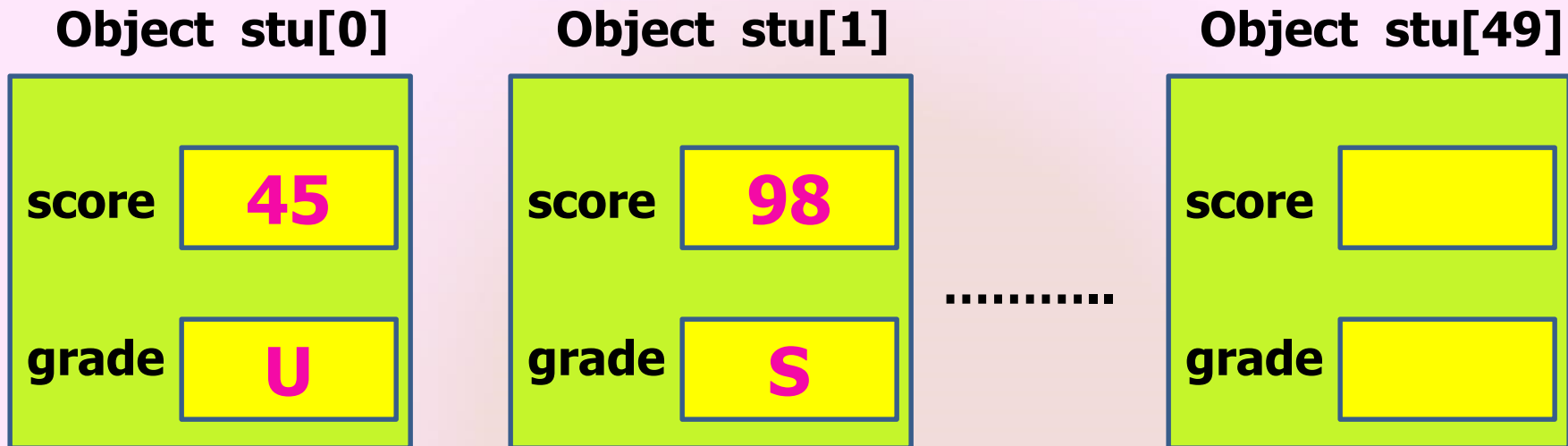
วิเคราะห์โจทย์

- ❁ การนิยามคลาสนักศึกษา จะพิจารณาจากคุณลักษณะและพฤติกรรมของนักศึกษาหนึ่งคนใด ๆ เท่านั้น โดยคลาสนักศึกษาจะเป็นต้นแบบสำหรับนักศึกษาทุกคน ไม่ว่านักศึกษาคนใดก็ตามจะต้องมีคุณลักษณะและพฤติกรรมเช่นเดียวกันกับที่ได้นิยามคลาสไว้ ส่วนการสร้างวัตถุจะสร้างตามจำนวนที่โจทย์กำหนด เช่นตัวอย่างนี้จะมีการสร้างวัตถุขึ้นมา N วัตถุ สำหรับใช้แทนนักศึกษา N คน
- ❁ ดังนั้นการนิยามคลาสนักศึกษา (Class Student) จะประกอบด้วย
 - ข้อมูลสมาชิก คือ คะแนน (score) และ เกรด (grade)
 - เมทอดสมาชิก คือ การรับคะแนน (setScore) การตัดเกรด (calGrade) และการพิมพ์เกรด (print Grade)

```
import java.util.Scanner;
class Student {           //class definition
    private int score;
    private char grade;
    public void setScore(int i) {           //method definition
        Scanner inVar = new Scanner(System.in);
        System.out.print("Enter score of student no. "+ i +":");
        score = inVar.nextInt();
    }
    public void calGrade() { //method definition
        if (score < 50)
            grade = 'U' ;
        else
            grade = 'S' ;
    }
    public void printGrade(int i) {
        System.out.println("Score of student no. "+ i + " is " + score);
        System.out.println("Grade of student no. "+ i + " is " + grade + "\n");
    }
} //end class Student
```

```
public class JavaApp3 {  
    public static void main(String[] args) {  
  
        Scanner inVar = new Scanner(System.in);  
        int i, N;  
        System.out.print("Enter N : ") ;  
        N = inVar.nextInt();  
  
        Student [] stu = new Student [N];  
        for (i=0;i<N;i++) {  
            stu[i] = new Student();  
            stu[i].setScore(i+1);  
            stu[i].calGrade();  
            stu[i].printGrade(i+1);  
        }  
    }  
} //end class JavaApp3
```

ตัวอย่างผลลัพธ์



Enter N : 10
Enter score of student no. 1: 45
Score of studen no. 1 is 45
Grade of studen no. 1 is U

Enter score of student no. 2: 98
Score of studen no. 2 is 98
Grade of studen no. 2 is S

.....
Enter score of student no. 10: 80
Score of studen no. 2 is 80
Grade of studen no. 2 is S