



ใบความรู้รายวิชา การวิเคราะห์ข้อมูล
ระดับประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)
หน่วยที่ 5 : การสร้างวัตถุ (Object)

หัวข้อการเรียนรู้

- 5.1 การกำหนดคลาส (Defining Class)
- 5.2 การสร้างวัตถุ (Object Instantiation)
- 5.3 การกำหนดค่าและการเรียกใช้เมทอด (Setting Attributes and Calling Methods)
- 5.4 การเข้าถึงและการประมวลผลข้อมูล (Accessing and Processing Data)
- 5.5 การทำงานร่วมกับวัตถุอื่น (Working with Other Objects)
- 5.6 การทำงานกับคลาสอื่น (Inheritance)
- 5.7 การทำงานกับอินสแตนซ์ของคลาส (Instance Operations)

5.1 การกำหนดคลาส (Class Definition)

คลาส (Class) คือ แม่แบบ (Template) หรือโครงสร้างที่ใช้ในการสร้างวัตถุ (Object) ซึ่งประกอบด้วย

- แอตทริบิวต์ (Attributes) : ตัวแปรที่เก็บข้อมูลของวัตถุ
- เมทอด (Methods) : ฟังก์ชันที่ใช้กำหนดพฤติกรรมของวัตถุ

◆ ตัวอย่าง (ภาษา Python)

class Student:

```
def __init__(self, name, score):  
    self.name = name  
    self.score = score
```

```
def show_info(self):  
    print(f"ชื่อนักเรียน: {self.name}, คะแนน: {self.score}")
```

5.2 การสร้างวัตถุ (Object Instantiation)

การสร้างวัตถุ คือ การนำคลาสไปใช้งานจริง โดยการประกาศอินสแตนซ์จากคลาสนั้น

◆ ตัวอย่าง

```
stu1 = Student("อนันต์", 85)
```

```
stu2 = Student("ศิริพร", 90)
```

วัตถุ stu1 และ stu2 เป็นอินสแตนซ์ของคลาส Student

5.3 การกำหนดค่าและการเรียกใช้เมทอด (Attributes and Methods)

สามารถเข้าถึงหรือเปลี่ยนแปลงค่าภายในวัตถุได้โดยใช้เครื่องหมายจุด (.)

◆ ตัวอย่าง

```
stu1.score = 95          # เปลี่ยนค่าคะแนน
```

```
stu1.show_info()        # เรียกใช้เมทอด
```

5.4 การเข้าถึงและการประมวลผลข้อมูล (Accessing and Processing Data)

สามารถใช้ข้อมูลในวัตถุเพื่อประมวลผล หรือแสดงผลได้ตามต้องการ

◆ ตัวอย่าง

```
average = (stu1.score + stu2.score) / 2
print("คะแนนเฉลี่ย:", average)
```

5.5 การทำงานร่วมกับวัตถุอื่น (Working with Other Objects)

วัตถุสามารถทำงานร่วมกับวัตถุอื่นได้ เช่น การส่งวัตถุหนึ่งไปเป็นพารามิเตอร์ในเมทอดของอีกวัตถุหนึ่ง

◆ ตัวอย่าง

```
class Teacher:
    def evaluate(self, student):
        print(f'{student.name} ได้คะแนน {student.score}')

teacher1 = Teacher()
teacher1.evaluate(stu1)
```

5.6 การทำงานกับคลาสอื่น (Inheritance)

Inheritance (การสืบทอดคุณสมบัติ) คือ การสร้างคลาสใหม่โดยอ้างอิงจากคลาสเดิม เพื่อขยายหรือปรับปรุงการทำงาน

◆ ตัวอย่าง

```
class GraduateStudent(Student):
    def thesis(self, topic):
        print(f'{self.name} ทำวิจัยเรื่อง {topic}')

grad1 = GraduateStudent("ปริญญา", 88)
grad1.show_info()
grad1.thesis("การวิเคราะห์ข้อมูลทางธุรกิจ")
```

5.7 การทำงานกับอินสแตนซ์ของคลาส (Instance Operations)

สามารถสร้างอินสแตนซ์หลายตัวจากคลาสเดียวกัน และจัดการข้อมูลของแต่ละอินสแตนซ์แยกจากกันได้

◆ ตัวอย่าง

```
students = [
    Student("สุชาติ", 78),
    Student("วีระ", 82),
    Student("มนิรัตน์", 91)
]
```

```
for s in students:
    s.show_info()
```