

บทที่ 8

การเขียนโปรแกรมเชิงวัตถุเบื้องต้น

Chapter 8 Introduction to
Object-Oriented Programming: OOP in Python



Asst.Prof.Dr.Nutthapat Kaewrattanapat
Suan Sunandha Rajabhat University

Course: Computer Programming and Developing Applications for Education
Department of Digital Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University

อาจารย์บรรยาย



ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร

การศึกษา: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

- 2565 ปริญญาเอก ปรัชญาดุสิตบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการสื่อสารเพื่อการศึกษา (GPA. 4.00)
- 2551 ปริญญาโท วิทยาศาสตร์มหาบัณฑิต สาขาวิชาระบบสารสนเทศเพื่อการจัดการ (GPA. 3.58)
- 2549 ปริญญาตรี วิทยาศาสตร์บัณฑิต เกียรตินิยมอันดับ 1 สาขาวิชาวิทยาการคอมพิวเตอร์ (GPA. 3.66)

คุณวุฒิวิชาชีพและประกาศนียบัตร

- คุณวุฒิวิชาชีพ จาก สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน) สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขาธุรกิจดิจิทัลและพาณิชย์อิเล็กทรอนิกส์ อาชีพนักจัดการเทคโนโลยีสารสนเทศสำหรับธุรกิจ ระดับ 6 เลขที่หนังสือรับรอง PQCNI-ICT-ECM-0-251100-B-64/000029
- วิทยาศาสตร์ข้อมูลด้วยภาษาไพทอน, มหาวิทยาลัยธรรมศาสตร์
- การโปรแกรมสำหรับนักภาษาศาสตร์, มหาวิทยาลัยมหิดล
- การโปรแกรมภาษาไพทอน, มหาวิทยาลัยเพนซิลวาเนีย, สหรัฐอเมริกา
- การโปรแกรมสำหรับทุกคน, มหาวิทยาลัยมิชิแกน, สหรัฐอเมริกา

ติดต่อ: nutthapat.ke@ssru.ac.th

Course Description

หลักการ ทฤษฎี ที่เกี่ยวข้องกับการเขียนโปรแกรมและพัฒนาแอปพลิเคชัน หลักการ พัฒนาโปรแกรมคอมพิวเตอร์ คุณสมบัติของโปรแกรมภาษาชนิดต่าง ๆ หลักการ เบื้องต้นเกี่ยวกับองค์ประกอบ ลักษณะคำสั่ง การเขียนโปรแกรม ขั้นตอนวิธี การ วิเคราะห์ การออกแบบ แอปพลิเคชันเพื่อการศึกษา การประเมินซอฟต์แวร์ สามารถพัฒนาแอปพลิเคชันเพื่อการศึกษา

Principles, theories associated with computer programming and development applications, computer programming principles, computer language, elements of computer language, syntax, computer programming, algorithms, analysis and design application for education, software evaluation, candidate teachers able to developing applications for education.

Algorithm Design

Python Programming Language

System Development Life Cycle: SDLC

Reference: <https://edu.ssru.ac.th/useruploads/files/20230724/1772131ed638786bc8d19918b37249af72c36be4.pdf>

Course Outline

- Chapter 1 – Computer Programming
- Chapter 2 – Introduction to Python Programming
- Chapter 3 – Conditional Statement
- Chapter 4 – Iteration Statement
- Chapter 5 – Strings
- Chapter 6 – Immutable and Mutable Data Types
- Chapter 7 – Functions
- Chapter 8 – Object-Oriented Programming: OOP
- Chapter 9 – Python and Database Management System
- Chapter 10 – System Development Life Cycle: SDLC

Measurement and Evaluation

การวัดและประเมินผล

1. ระหว่างการจัดการเรียนรู้

- สอบ Pre-test
- การมอบหมายงาน
- สอบ Post-test
- การมีส่วนร่วมในชั้นเรียน

0%
24%
12%
4%

2. การสอบกลางภาค (Midterm Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

3. โครงการประจำภาคเรียน (Term Project)

- โครงการและการนำเสนอ

20%

4. การสอบปลายภาค (Final Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

ร้อยละ	ระดับผลการเรียน	ความหมาย
86 – 100	A	ดีเยี่ยม
82 – 85	A-	ดีเยี่ยม
78 – 81	B+	ดีมาก
74 – 77	B	ดี
70 – 73	B-	ค่อนข้างดี
66 – 69	C+	ปานกลางค่อนข้างดี
62 – 65	C	ปานกลาง
58 – 61	C-	ปานกลางค่อนข้างอ่อน
54 – 57	D+	ค่อนข้างอ่อน
50 – 53	D	อ่อน
46 – 49	D-	อ่อนมาก
0 – 45	F	ตก

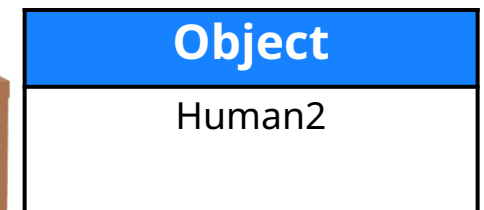
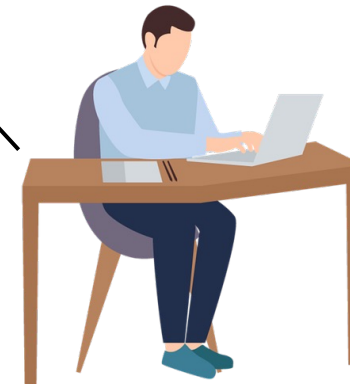
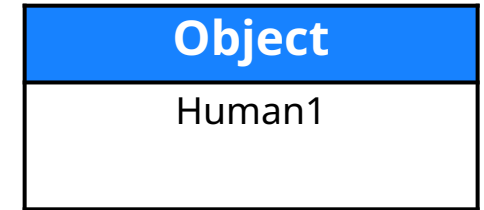
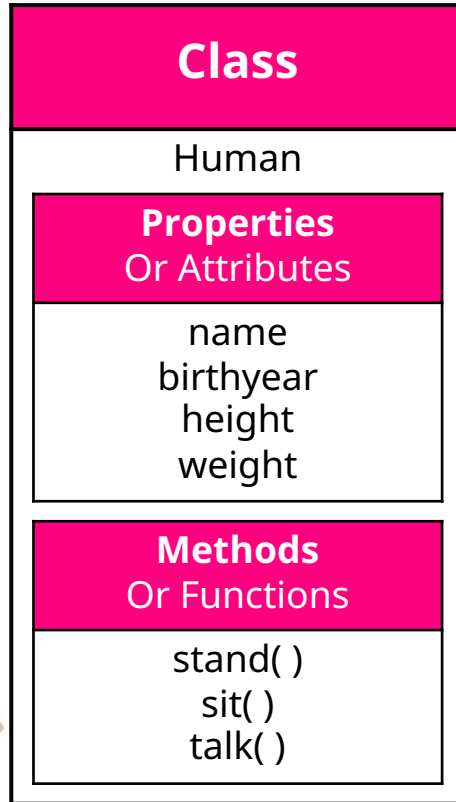
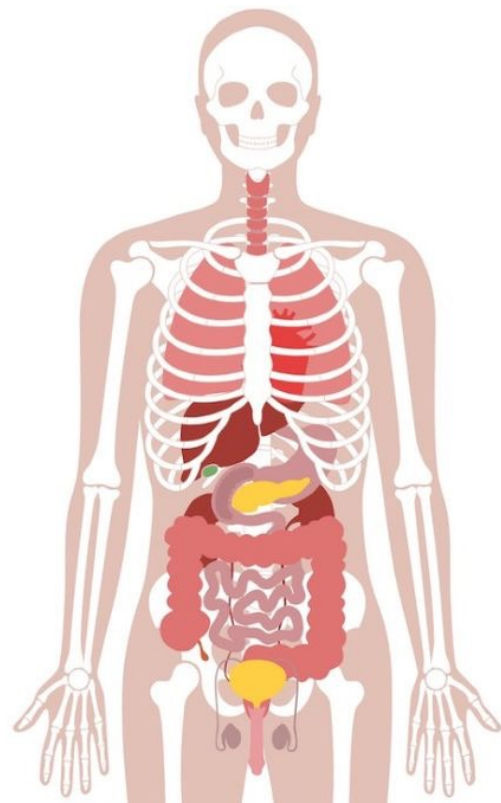
Measurement and Evaluation

ครั้งที่/วันที่	บทเรียน/หัวข้อ	มอบหมายงาน (24%)	สอบ Post-test (12%)	การมีส่วนร่วมในชั้นเรียน (4%)
1 พท 7 ธค 66	แนะนำรายวิชา การวัดและการประเมินผล หัวข้อเรียนรู้ บทที่ 1 การโปรแกรมคอมพิวเตอร์	-	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
2 พท 14 ธค 66	บทที่ 2 พื้นฐานการโปรแกรมภาษาไพทอน	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
3 พท 21 ธค 66	บทที่ 3 การโปรแกรมแบบเงื่อนไข (Conditional Statement)	-	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
4 พท 28 ธค 66 Online	บทที่ 3 การโปรแกรมแบบเงื่อนไข (Conditional Statement) (ต่อ)	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
5 พท 4 มค 67	บทที่ 4 การโปรแกรมแบบทำซ้ำ (Iteration Statement)	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
6 พท 11 มค 67	บทที่ 5 การโปรแกรมสายอักขระ (String)	-	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
7 จ 22 มค 67	สอบกลางภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%			
8 พท 1 กพ 67	บทที่ 6 ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types) : Tuple, List และ Dictionary	6 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%

Measurement and Evaluation

ครั้งที่/วันที่	บทเรียน/หัวข้อ	มอบหมายงาน (24%)	สอบ Post-test (12%)	การมีส่วนร่วมในชั้นเรียน (4%)
9 พฤษ 8 กพ 67	บทที่ 7 ฟังก์ชัน	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
10 พฤษ 15 กพ 67	บทที่ 8 การโปรแกรมเชิงวัตถุ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
12 พฤษ 22 กพ 67	บทที่ 9 ภาษาไพทอนกับระบบการจัดการฐานข้อมูล	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
13 พฤษ 29 กพ 67	บทที่ 9 ภาษาไพทอนกับระบบการจัดการฐานข้อมูล ต่อ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
14 พฤษ 7 มีค 67	บทที่ 10 วงจรการพัฒนาระบบ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
15 พฤษ 14 มีค 67	สอบปลายภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%			
16 พฤษ 21 มีค 67	ส่งโครงงาน และนำเสนอ 20%			

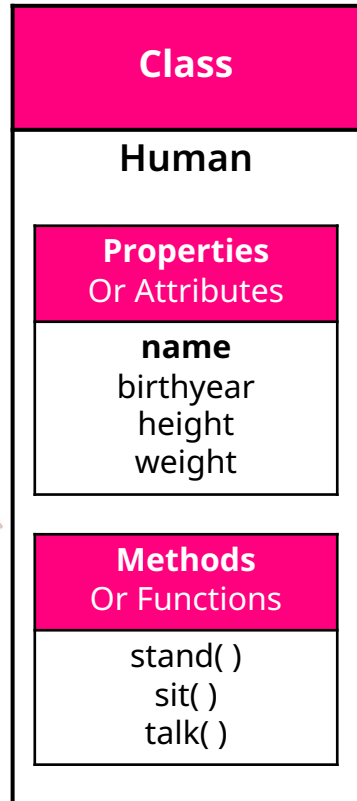
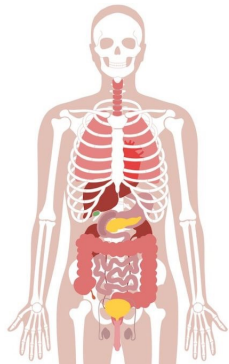
Object-Oriented Programming: OOP



<https://www.livescience.com/37009-human-body.html>

Object-Oriented Programming: OOP

Namespace: Human



```
class Human:
    def __init__(self, name=None, birth_year=None, height=None, weight=None):
        self.name = name
        self.birthyear = birth_year
        self.height = height
        self.weight = weight
```

Object/Instance: human1

```
human1 = Human(name="Nutthapat")
print(human1.name)
```

Object/Instance: human2

```
human2 = Human(name="Mongkolchai")
print(human2.name)
```

คลาส (Class)

- ใน Python นั้น class คือ คำสำคัญที่ใช้สร้าง "คลาส (Class)" ซึ่งเป็นแบบแผนหรือโครงสร้างสำหรับสร้างอ็อบเจกต์ (Objects) หรืออินสแตนท์ (Instance)
- คลาสเป็นแบบแผนหรือโมเดลที่อธิบายถึงลักษณะและพฤติกรรมของอ็อบเจกต์ที่จะถูกสร้างขึ้น ซึ่งประกอบด้วยแอตทริบิวต์หรือพร็อพเพอร์ตี้ (attributes/properties) และเมธอด (methods)

```
class Human:
    def __init__(self, name=None, birth_year=None, height=None, weight=None):
        self.name = name
        self.birthyear = birth_year
        self.height = height
        self.weight = weight
```

วัตถุ (Object)

- class เป็นต้นแบบเท่านั้น ไม่สามารถทำงานได้ จนกว่าจะมีการสร้าง object ที่มีคุณลักษณะเหมือนกับต้นแบบ class นั้นๆ ขึ้นมา
- ซึ่งการมี class เป็นต้นแบบนี้ ทำให้สามารถสร้าง object ได้มากกว่า 1 object เปรียบเสมือนการมีแบบแปลนบ้าน ทำให้สร้างบ้านได้หลายๆ กันหลายหลัง หรือการมีแปลนรถยนต์ก็ทำให้สร้างรถยนต์ที่มีความเหมือนกันได้หลายคัน



```
class Human:
    def __init__(self, name=None, birth_year=None, height=None, weight=None):
        self.name = name
        self.birthyear = birth_year
        self.height = height
        self.weight = weight
```

```
human1 = Human(name="Nutthapat")
print(human1.name)
```

วัตถุ (Object)

- class เป็นต้นแบบเท่านั้น ไม่สามารถทำงานได้ จนกว่าจะมีการสร้าง object ที่มีคุณลักษณะเหมือนกับต้นแบบ class นั้นๆ ขึ้นมา
- ซึ่งการมี class เป็นต้นแบบนี้ ทำให้สามารถสร้าง object ได้มากกว่า 1 object เปรียบเสมือนการมีแบบแปลนบ้าน ทำให้สร้างบ้านได้หลายๆ กันหลายหลัง หรือการมีแปลนรถยนต์ก็ทำให้สร้างรถยนต์ที่มีความเหมือนกันได้หลายคัน



การเขียนโปรแกรมคอมพิวเตอร์และพัฒนาแอปพลิเคชันเพื่อการศึกษา
ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร์

```
class Human:
    def __init__(self, name=None, birth_year=None, height=None, weight=None):
        self.name = name
        self.birthyear = birth_year
        self.height = height
        self.weight = weight
```

```
human1 = Human(name="Nutthapat")
print(human1.name)
```

```
human2 = Human(name="Mongkolchai")
print(human2.name)
```

วัตถุ (Object)

- class เป็นต้นแบบเท่านั้น ไม่สามารถทำงานได้ จนกว่าจะมีการสร้าง object ที่มีคุณลักษณะเหมือนกับต้นแบบ class นั้นๆ ขึ้นมา
- ซึ่งการมี class เป็นต้นแบบนี้ ทำให้สามารถสร้าง object ได้มากกว่า 1 object เปรียบเสมือนการมีแบบแปลนบ้าน ทำให้สร้างบ้านได้หลายๆ กันหลายหลัง หรือการมีแปลนรถยนต์ก็ทำให้สร้างรถยนต์ที่มีความเหมือนกันได้หลายคัน



การเขียนโปรแกรมคอมพิวเตอร์และพัฒนาแอปพลิเคชันเพื่อการศึกษา
ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร์

```
class Human:
    def __init__(self, name=None, birth_year=None, height=None, weight=None):
        self.name = name
        self.birthyear = birth_year
        self.height = height
        self.weight = weight
```

```
human1 = Human(name="Nutthapat")
print(human1.name)
```

```
human2 = Human(name="Mongkolchai")
print(human2.name)
```

```
human3 = Human(name="Wichuda")
print(human3.name)
```

คอนสตรัคเตอร์ หรือ ผู้สร้าง (Constructor)

- Constructor มีลำดับการทำงานลำดับแรกในการสร้าง object ขึ้นมา
- โดย Constructor ช่วยให้สามารถกำหนดข้อมูลพื้นฐานให้กับตัวแปรหรือ properties ต่างๆ ได้
- Constructor ในภาษา python จะใช้คำว่า `def __init__()` ในการสร้าง

```
1 class Car:
2     def __init__(self, brand=None, color=None):
3         self.brand = brand
4         self.color = color
5
6
7
8 car1 = Car("Toyota", "Red")
9 print(car1.brand)
```

Toyota

การกำหนดการแสดงผล Object ด้วย __str__()

- __str__() เป็นเมธอดพิเศษ (special method) ใน Python ที่ถูกใช้ในการกำหนดลักษณะการแสดงผลของ Object เมื่อถูกแปลงเป็นสตริง (string representation)

```
1 class Car:
2     def __init__(self, brand=None, color=None):
3         self.brand = brand
4         self.color = color
5
6     def __str__(self):
7         return f"Car's brand {self.brand}, color {self.color}"
8
9
10 car1 = Car("Toyota", "Red")
11 print(car1.brand)
12 print(car1.color)
13 print(car1)
```

```
Toyota
Red
Car's brand Toyota, color Red
```

เมธอด (Methods)

- เมธอด คือ ฟังก์ชันการทำงานของ Class และ Object

```
1 import datetime
2 class Human:
3     def __init__(self, name=None, birth_year=None, height=None, weight=None):
4         self.name = name
5         self.birthyear = birth_year
6         self.height = height
7         self.weight = weight
8
9     def __str__(self):
10        return f"Name: {self.name} Birth Year: {self.birthyear} Height: {self.height} and Weight: {self.weight}"
11
12    def age(self):
13        today = datetime.date.today()
14        self.age = today.year - self.birthyear
15        return self.age
16
17
18 human1 = Human(name="Nutthapat", birth_year=1983)
19 print(human1.age())
```


การสืบทอด (Inheritance)

```
1 # การสืบทอด Inheritance
2 import datetime
3 class Human:
4     def __init__(self, name=None, birth_year=None, height=None, weight=None):
5         self.name = name
6         self.birthyear = birth_year
7         self.height = height
8         self.weight = weight
9
10    def __str__(self):
11        return f"Name: {self.name} Birth Year: {self.birthyear} Height: {self.height} and Weight: {self.weight}"
12
13    def age(self):
14        today = datetime.date.today()
15        self.age = today.year - self.birthyear
16        return self.age
17
18 # Teacher ได้รับการสืบทอดจาก Human
19 class Teacher(Human):
20     pass
21
22 teacher1 = Teacher(name="Nutthapat", birth_year=1983)
23 print(teacher1)
24
25
```

Name: Nutthapat Birth Year: 1983 Height: None and Weight: None

การสืบทอด (Inheritance)

```
1 # การสืบทอด Inheritance
2 import datetime
3 class Human:
4     def __init__(self, name=None, birth_year=None, height=None, weight=None):
5         self.name = name
6         self.birthyear = birth_year
7         self.height = height
8         self.weight = weight
9
10    def __str__(self):
11        return f"Name: {self.name} Birth Year: {self.birthyear} Height: {self.height} and Weight: {self.weight}"
12
13    def age(self):
14        today = datetime.date.today()
15        self.age = today.year - self.birthyear
16        return self.age
17
18 # Teacher ได้รับการสืบทอดจาก Human
19 class Teacher(Human):
20     def teaching(self, subject):
21         self.teachsubject = subject
22
23
24 teacher1 = Teacher(name="Nutthapat", birth_year=1983)
25 print(teacher1)
26 teacher1.teaching("Computer")
27 print(teacher1.teachsubject)
28
29
```

Name: Nutthapat Birth Year: 1983 Height: None and Weight: None
Computer

Assignment unit 8

1. ให้เขียนโปรแกรมเพื่อสร้าง Class สำหรับการหาพื้นที่ดังนี้

- กำหนดให้ชื่อคลาส คือ area
- สร้าง method ชื่อ `rectangle_area(self, w, h)` สำหรับการคำนวณหาพื้นที่สี่เหลี่ยมและส่งค่าพื้นที่สี่เหลี่ยมกลับไป (return)
- สร้าง method ชื่อ `circle_area(self, r)` สำหรับการคำนวณหาพื้นที่วงกลมและส่งค่าพื้นที่วงกลมกลับไป (return)
- สร้าง method ชื่อ `triangle_area(self, b, h)` สำหรับการคำนวณหาพื้นที่สามเหลี่ยมและส่งค่าพื้นที่สามเหลี่ยมกลับไป (return)
- สร้าง constructor เพื่อรับค่าชื่อรูปทรง (shapename) และหากชื่อรูปทรงตรงกับเงื่อนไขรูปทรงใด ให้เรียกใช้ method ที่ตรงกันและคำนวณหาพื้นที่

Assignment unit 8

2. ให้เขียนโปรแกรมเพื่อสร้าง Class สำหรับการแปลงอุณหภูมิ เซลเซียส-ฟาเรนไฮต์ และ ฟาเรนไฮต์-เซลเซียส ดังนี้

- กำหนดให้ชื่อคลาส คือ temperature
- สร้าง method ชื่อ cel_to_far(self, celcius) สำหรับการแปลงเซลเซียสเป็นฟาเรนไฮต์และส่งค่ากลับไป (return)
- สร้าง method ชื่อ far_to_cel(self, far) สำหรับการแปลงฟาเรนไฮต์เป็นเซลเซียสและส่งค่ากลับไป
- สร้าง constructor โดยไม่ต้องรับค่าใดๆ
- สร้าง __str__ สำหรับการแสดงข้อความรายละเอียดของ class temperature ดังนี้ "class temperature คือ คลาส สำหรับการแปลงอุณหภูมิ เซลเซียส-ฟาเรนไฮต์ cel_to_far และ ฟาเรนไฮต์-เซลเซียส far_to_cel"