

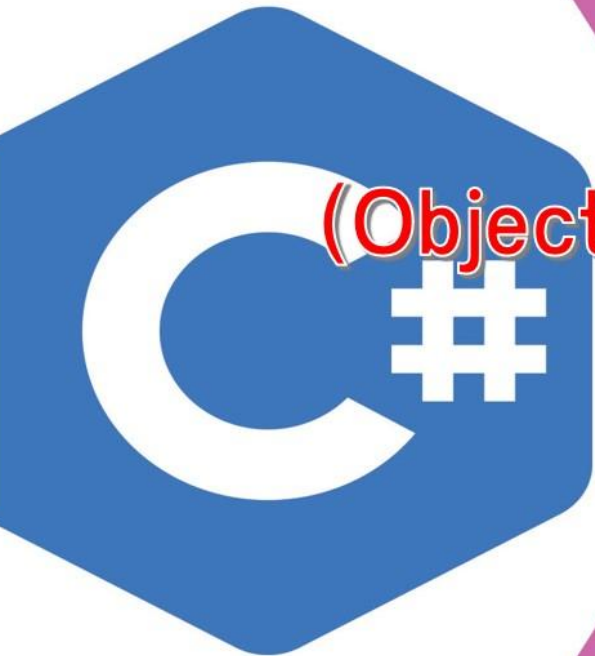
รหัสวิชา 20204-2108



# การเขียนโปรแกรม

## เชิงวัตถุเบื้องต้น

(Object-Oriented Programming Basic )



โดย

วราภรณ์ อุ่มอังวะ



# หน่วยที่ 1

## บทบาทของโปรแกรม

## เชิงวัตถุเบื้องต้น





# หัวข้อเรื่อง

1.1 ความหมายของโปรแกรมเชิงวัตถุ

1.2 บทบาทของโปรแกรมเชิงวัตถุ

1.3 ความเป็นมาและแนวคิดการเขียน  
โปรแกรมเชิงวัตถุ

1.4 คุณสมบัติของโปรแกรมเชิงวัตถุ

1.5 ประโยชน์ของโปรแกรมเชิงวัตถุ



## 1.1

### ความหมายของโปรแกรมเชิงวัตถุ

**การทำโปรแกรมเชิงวัตถุ** หมายถึง วิธีการเขียนโปรแกรมของนักเขียนโปรแกรมรุ่นใหม่ ที่จัดแบ่งการเขียนคำสั่งกันออกเป็นชุด ๆ แต่ละชุดเรียกว่า "วัตถุ" (object) แล้วจึงนำเอาชุดคำสั่งแต่ละชุดนั้นมารวมกันเป็นโปรแกรมชุดใหญ่อีกทีหนึ่ง ในบางครั้งยังอาจนำ "วัตถุ" ของโปรแกรมหนึ่งไปรวมกับ "วัตถุ" ของอีกโปรแกรมหนึ่ง แล้วเรียกออกมาใช้ได้เลย



## 1.2

## บทบาทของโปรแกรมเชิงวัตถุ

**โปรแกรมเชิงวัตถุ OOP** เป็นวิธีการเขียนโปรแกรม โดยอาศัยแนวคิดของวัตถุชิ้นหนึ่ง มีความสามารถในการปกป้องข้อมูล และการสืบทอดคุณสมบัติ ซึ่งทำให้แนวโน้มของOOP ได้รับการยอมรับและได้พัฒนามาใช้ในระบบต่าง ๆ มากมาย เช่น ระบบปฏิบัติการวินโดวส์ เป็นต้น

การเขียนโปรแกรมเชิงวัตถุ OOP : Object Oriented Programming คือหนึ่งในรูปแบบการเขียนโปรแกรมคอมพิวเตอร์ ที่ให้ความสำคัญกับวัตถุ





### 1.3

## ความเป็นมาและแนวคิดการเขียนโปรแกรมเชิงวัตถุ

**แนวคิดดั้งเดิมของการเขียนโปรแกรม** ก็คือ การแก้ปัญหาโดยใช้คอมพิวเตอร์เป็นเครื่องมือ แนวคิดนี้คล้ายกับการใช้เครื่องคิดเลขในการแก้ปัญหาทางคณิตศาสตร์

**แนวคิดแบบใหม่ที่ใช้ในการการเขียนโปรแกรม** ก็คือ การเน้นถึงปัญหาและองค์ประกอบของปัญหาเพื่อแก้ปัญหา การเน้นที่ปัญหาและองค์ประกอบของการแก้ปัญหา จะคล้ายกับการแก้ไขปัญหาและชีวิตความเป็นอยู่ของมนุษย์ที่จะต้อง มี คน สัตว์ สิ่งของ เพื่อแก้ปัญหา

อาลัน เคย์ (Alan Kay) เป็นผู้บุกเบิกแนวความคิดในการเขียนโปรแกรมเชิงวัตถุ ซึ่งเป็นต้นแบบของการเขียนโปรแกรมเชิงวัตถุได้เสนอกฎ 5 ข้อ ที่เป็นแนวทางของภาษาคอมพิวเตอร์เชิงวัตถุไว้ดังนี้

1. ทุก ๆ สิ่งเป็นวัตถุ
2. โปรแกรม ก็คือ กลุ่มของวัตถุที่ส่งข่าวสารบอกกันและกันให้ทำงาน
3. ในวัตถุแต่ละวัตถุจะต้องมีหน่วยความจำและประกอบไปด้วยวัตถุอื่นๆ
4. วัตถุทุกชนิดจะต้องจัดอยู่ในประเภทใดประเภทหนึ่ง
5. วัตถุที่จัดอยู่ในประเภทเดียวกันย่อมได้รับข่าวสารเหมือนกัน

### แนวคิดแบบ OOP

ถ้าเราไม่มองในแง่มุมของการเขียนโปรแกรมเพียงอย่างเดียว ให้มองไปในภาพรวม มองไปในสิ่งรอบ ๆ ตัวเราสามารถบอกได้ว่าแนวคิดของ OOP ก็คือ “ธรรมชาติของวัตถุ” หมายความว่า OOP จะมองสิ่งแต่ละสิ่งถือเป็น “วัตถุชิ้นหนึ่ง”

เนื่องจากหลักการเขียนโปรแกรมเชิงวัตถุเป็นแนวคิดแบบใหม่ ดังนั้น การทำงานหลาย ๆ ส่วนของการเขียนโปรแกรมแบบนี้ ซึ่งยังไม่เป็นที่คุ้นเคยมากนัก จึงควรมีความรู้เบื้องต้นเกี่ยวกับ OOP ดังนี้

การเชื่อมต่อ

การซ่อนรายละเอียด

การนำวัตถุมาใช้ใหม่

การพ้องรูป

การเขียนโปรแกรมและออกแบบระบบงาน

เปรียบเทียบแนวคิดระหว่างการเขียนโปรแกรมเชิงกระบวนการและเชิงวัตถุ

แนวทางการออกแบบและแก้ปัญหา

ดีไซน์แพตเทิร์น-แบบแผนและแนวทางการออกแบบ

การเขียนโปรแกรมเชิงวัตถุและฐานข้อมูล

โปรแกรมเชิงวัตถุและการเทียบเคียงกับโลกของความเป็นจริง



1.4

คุณสมบัติของโปรแกรมเชิงวัตถุ

1. ความสามารถในการสืบทอด (Inheritance)

2. ความสามารถในการเก็บซ่อน (Encapsulation)

3. ความสามารถในการแปลงร่าง (Polymorphism)

4. ความสามารถในการจัดการโครงสร้างข้อมูลแบบ  
เชิงนามธรรม (Abstract)

## 1.5

### ประโยชน์ของโปรแกรมเชิงวัตถุ

**1. ความสามารถในการเรียกใช้ได้หลายครั้ง** ออบเจกต์ได้ถูกออกแบบตามหลักการที่ว่าสามารถเรียกใช้งานได้หลาย ๆ ครั้ง ในหลักการนี้ทำให้แอปพลิเคชัน ของ OOP ตัวแรกอาจจะทำได้ยาก แต่ว่าโปรแกรมแอปพลิเคชันที่เขียนภายหลังจะสร้างง่ายเพราะสามารถเรียกใช้ออบเจกต์ที่ถูกสร้างไว้ตั้งแต่โครงงานแรกได้

**2. ความเชื่อถือได้** โปรแกรมแอปพลิเคชันของ OOP จะมีความเชื่อถือได้สูง เพราะจะรวมเอาส่วนย่อยที่ทดสอบจนได้มาตรฐานแล้วมารวมเข้าไว้ด้วยกัน รหัส (Code) ที่เขียนขึ้นมาใหม่ในแต่ละแอปพลิเคชันจะมีไม่มากนัก

**3. ความต่อเนื่องกัน** การพัฒนาซอฟต์แวร์แบบ OOP ใน C++ จะเปลี่ยนไปตามฝีมือและจำนวนนักเขียนโปรแกรมภาษา C นักเขียนโปรแกรมภาษาซีสามารถรู้หลักการของ OOP ได้ภายในเวลาไม่นาน และสามารถเข้าใจเนื้อหาได้ไม่ยาก อีกทั้งสามารถแปลงโปรแกรมแอปพลิเคชันของ C เป็น C++ ได้อย่างรวดเร็ว