

CHAPTER02

การวิเคราะห์ปัญหา และการออกแบบโปรแกรม

บทนำ

- ❑ ก่อนการเขียนโปรแกรมจำเป็นที่จะต้องทำการวิเคราะห์ปัญหา และออกแบบโปรแกรม เพราะจะช่วยทำให้กระบวนการทางความคิดของโปรแกรมเมอร์มีลำดับขั้นตอน
- ❑ นอกจากนี้ก็ยังช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นในขณะที่เขียนโปรแกรมได้

การวิเคราะห์โจทย์ปัญหา

- สำหรับการวิเคราะห์โจทย์ปัญหานั้น เมื่อเราอ่านโจทย์ปัญหาเสร็จ ให้เราพิจารณาว่า โจทย์ปัญหาต้องการอะไร โจทย์ปัญหากำหนดอะไรมาบ้าง และการได้มาซึ่งความต้องการของโจทย์จะได้มาอย่างไร
- เขียนเป็น **block diagram** การวิเคราะห์โจทย์ปัญหาได้ ดังนี้



การวิเคราะห์โจทย์ปัญหา (ต่อ)

- ❑ จาก **block diagram**
- ❑ **INPUT** พิจารณาว่าข้อมูลใดที่ต้องการนำเข้าสู่โปรแกรม
- ❑ **PROCESS** เป็นขั้นตอนการประมวลผลตามลำดับเหตุการณ์ โดยขั้นตอนนี้จะถูกนำไปเขียนอัลกอริทึมในการออกแบบโปรแกรมต่อไป
- ❑ **OUTPUT** คือสิ่งที่โจทย์ปัญหาต้องการ หรือผลลัพธ์นั่นเอง

ตัวอย่างการวิเคราะห์โจทย์ปัญหา

ตัวอย่างที่ 1 ให้รับค่าตัวเลข **3** จำนวนจากแป้นพิมพ์ แล้วหาผลรวม และค่าเฉลี่ยของตัวเลขทั้ง **3** แสดงออกทางจอภาพ

จากโจทย์พิจารณาแล้วแบ่งออกเป็นได้ **3** ส่วน ดังนี้

INPUT	PROCESS	OUTPUT
ตัวเลข 3 จำนวน	รับค่าตัวเลขมา 3 จำนวน	ผลรวมของตัวเลขทั้ง 3
	หาผลรวมของตัวเลขทั้ง 3	ค่าเฉลี่ยของตัวเลขทั้ง 3
	หาค่าเฉลี่ยโดยนำผลรวมของตัวเลขทั้ง 3 มาหารด้วย 3	
	แสดงผลทางจอภาพ	

ตัวอย่างการวิเคราะห์โจทย์ปัญหา (ต่อ)

ตัวอย่างที่ 2 ให้คำนวณราคาขายสินค้าสุทธิ โดยกำหนดให้มีส่วนลด 5% แล้วแสดงผลทางจอภาพ

จากโจทย์พิจารณาแล้วแบ่งออกเป็นได้ 3 ส่วน ดังนี้

INPUT	PROCESS	OUTPUT
ราคาสินค้า	รับราคาสินค้า และจำนวนที่ซื้อ	ราคาขายสินค้าสุทธิ
จำนวน	คำนวณราคาขายสุทธิ ตามสูตร $(\text{ราคาสินค้า} * \text{จำนวน}) - [(\text{ราคาสินค้า} * \text{จำนวน}) * \% \text{ ส่วนลด}]$	
	แสดงผลทางจอภาพ	

การออกแบบโปรแกรม และการเขียนอัลกอริทึม

- ❑ จากขั้นตอนการวิเคราะห์ปัญหา นำมาสู่ขั้นตอนการออกแบบโปรแกรม
- ❑ ซึ่งนับว่าเป็นการวางแผนลำดับวิธีการทางคอมพิวเตอร์ หรือลำดับขั้นตอนของชุดคำสั่งโปรแกรมคอมพิวเตอร์ไว้ล่วงหน้า
- ❑ เราเรียกวิธีการนี้ว่า “อัลกอริทึม (Algorithm)”
- ❑ โดยอัลกอริทึมก็คือ ขั้นตอนของ **PROCESS** ที่นำข้อมูลเข้ามาเพื่อให้ได้ผลลัพธ์ที่ต้องการ นั่นเอง

การออกแบบโปรแกรม และการเขียนอัลกอริทึม (ต่อ)

- ในการออกแบบโปรแกรมจะต้องประกอบไปด้วย
- 1. การกำหนดรูปแบบผลลัพธ์ที่โจทย์ต้องการ (Output)
- 2. การกำหนดตัวแปรแทนข้อมูลที่คอมพิวเตอร์จะประมวลผล (Input)
- 3. เขียนอัลกอริทึม (Process)
- 4. การตรวจสอบข้อมูลที่นำเข้ามา เพื่อป้องกันข้อผิดพลาด (Process)

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

ตัวอย่างที่ 3 จากตัวอย่างที่ 1 ให้รับค่าตัวเลข 3 จำนวน แล้วหาผลรวมและค่าเฉลี่ยของตัวเลขทั้ง 3 แสดงผลทางจอภาพ

โดยกำหนดรูปแบบการแสดงผลล์พ์ดังนี้

Number 1 is :

Number 2 is :

Number 3 is :

Total Number is :

Average is :

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

1. ขั้นตอนการวิเคราะห์ปัญหา

จากโจทย์พิจารณาแล้วแบ่งออกได้เป็น 3 ส่วน ดังนี้

INPUT	PROCESS	OUTPUT
ตัวเลข 3 จำนวน	รับข้อมูลที่เป็นตัวเลข 3 จำนวนจากแป้นพิมพ์	ผลรวมตัวเลขทั้ง 3 จำนวน
	นำตัวเลขทั้ง 3 จำนวนมารวมกัน	ค่าเฉลี่ยของตัวเลขทั้ง 3
	นำผลรวมของตัวเลขทั้งสามมาหารด้วย 3	
	แสดงผลทางจอภาพ	

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

2. ขั้นตอนการออกแบบโปรแกรม

2.1 กำหนดตัวแปร

Num1	แทน	ตัวแปรจำนวนที่ 1
Num2	แทน	ตัวแปรจำนวนที่ 2
Num3	แทน	ตัวแปรจำนวนที่ 3
TotalNum	แทน	ตัวแปรผลรวมตัวเลขทั้ง 3 จำนวน
Average	แทน	ตัวแปรค่าเฉลี่ยตัวเลขทั้ง 3 จำนวน

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

2.2 อัลกอริทึม

1. รับค่าตัวเลข Num1, Num2, และ Num3 จากแป้นพิมพ์
2. $TotalNum = Num1 + Num2 + Num3$
3. $Average = TotalNum / 3$
4. พิมพ์ TotalNum และ Average
5. จบการทำงาน

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

ตัวอย่างที่ 4 จากตัวอย่างที่ 2 ให้คำนวณราคาขายสินค้าสุทธิ โดยกำหนดให้มีส่วนลด **5%** แล้วแสดงผลทางจอภาพ โดยกำหนดรูปแบบการแสดงผลลัพท์ดังนี้

คำนวณราคาขายสินค้าสุทธิ

กรุณป้อนราคาสินค้า :

จำนวนสินค้าที่ขาย :

ส่วนลด **5%** :

ราคาขายสุทธิ :

ต้องการทำงานต่อหรือไม่ [Y/N] :

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

1. ขั้นตอนการวิเคราะห์ปัญหา

จากโจทย์พิจารณาแล้วแบ่งออกเป็น 3 ส่วน ดังนี้

INPUT	PROCESS	OUTPUT
ราคาสินค้า	พิมพ์หัวข้อรายงาน	ราคาขายสินค้าสุทธิ
จำนวน	รับราคาสินค้า และจำนวนที่ซื้อ	
	คำนวณส่วนลด จากสูตร $(\text{ราคาสินค้า} * \text{จำนวน}) * 5\%$	
	คำนวณราคาขายสุทธิ ตามสูตร $(\text{ราคาสินค้า} * \text{จำนวน}) - \text{ส่วนลด}$	
	แสดงผลทางจอภาพ	

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

2. ขั้นตอนการออกแบบโปรแกรม

2.1 กำหนดตัวแปร

Price	แทน	ราคาขายสินค้า
Qty	แทน	จำนวนสินค้าที่ขาย
Discount	แทน	ส่วนลด
NetPrice	แทน	ราคาขายสุทธิ
Continue	แทน	ความต้องการทำงานต่อ
Y	แทน	ต้องการทำงานต่อ
N	แทน	ไม่ต้องการทำงานต่อ

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

2.2 อัลกอริทึม

1. รับค่า Price, Qty
2. $\text{Discount} = (\text{Price} * \text{Qty}) * 5\%$
3. $\text{NetPrice} = (\text{Price} * \text{Qty}) - \text{Discount}$
4. พิมพ์ NetPrice
5. ตรวจสอบ Continue
 - 5.1 หากเป็น Y คือ ต้องการทำงานต่อ ให้กลับไปทำงานข้อที่ 1 ใหม่
 - 5.2 หากเป็น N คือ ไม่ต้องการทำงานต่อ ให้ไปทำงานข้อที่ 6
6. จบการทำงาน

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

ตัวอย่างที่ 5 คำนวณหาเกรดจากคะแนนรวมทั้งหมดของนักเรียน โดยมีเงื่อนไขและกำหนดรูปแบบผลลัพธ์ ดังนี้

คะแนน	เกรด
80-100	A
70-79	B
60-69	C
50-59	D
0-49	F

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

รูปแบบการแสดงผลลัพธ์

เกรดนักเรียน

กรุณาป้อนชื่อ-นามสกุล :

คะแนนรวม :

เกรดที่ได้คือ :

ต้องการทำงานต่อหรือไม่ [Y/N] :

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

1. ขั้นตอนการวิเคราะห์ปัญหา

จากโจทย์พิจารณาแล้วแบ่งออกเป็น 3 ส่วน ดังนี้

Input	Process	Output
ชื่อ-สกุลนักเรียน	พิมพ์หัวข้อรายงาน	เกรด
คะแนนรวม	รับชื่อ-นามสกุล และคะแนน	
	ตรวจสอบเงื่อนไขคะแนนตามเงื่อนไข	
	พิมพ์ผลลัพธ์คือ เกรดที่ได้	

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

2. ขั้นตอนการออกแบบโปรแกรม

2.1 กำหนดตัวแปร

StdName	แทน	ชื่อ-นามสกุลนักเรียน
Score	แทน	คะแนนรวม
Grade	แทน	เกรดที่ได้
Continue	แทน	ตรวจสอบความต้องการทำงานต่อ
Y	แทน	ต้องการทำงานต่อ
N	แทน	ไม่ต้องการทำงานต่อ

ตัวอย่างการออกแบบโปรแกรม และการเขียนอัลกอริทึม

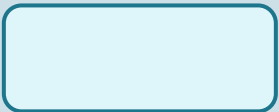
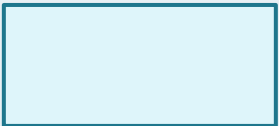
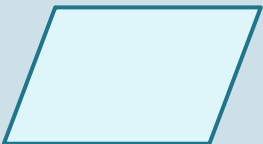
2.2 อัลกอริทึม

1. พิมพ์หัวข้อรายงาน
2. รับค่า StdName, Score
3. ถ้า Score น้อยกว่า 0 หรือมากกว่า 100 ให้ไปทำงานข้อ 5
4. ทดสอบค่า Score ตามเงื่อนไขดังนี้
 - 4.1 ถ้า $\text{Score} \geq 80$ พิมพ์ “เกรดที่ได้คือ : ” ; Grade = “A” แล้วไปทำข้อ 5
 - 4.2 ถ้า $\text{Score} \geq 70$ พิมพ์ “เกรดที่ได้คือ : ” ; Grade = “B” แล้วไปทำข้อ 5
 - 4.3 ถ้า $\text{Score} \geq 60$ พิมพ์ “เกรดที่ได้คือ : ” ; Grade = “C” แล้วไปทำข้อ 5
 - 4.4 ถ้า $\text{Score} \geq 50$ พิมพ์ “เกรดที่ได้คือ : ” ; Grade = “D” แล้วไปทำข้อ 5
 - 4.5 นอกเหนือจากเงื่อนไขข้างต้น ให้พิมพ์ “เกรดที่ได้คือ : ” ; Grade = “F” แล้วไปทำข้อ 5
5. ตรวจสอบค่า Continue
 - 5.1 หากเป็น Y คือ ต้องการทำงานต่อ ให้กลับไปทำงานตามข้อ 2
 - 5.2 หากเป็น N คือ ไม่ต้องการทำงานต่อ ให้ไปทำงานข้อ 6
6. จบการทำงาน

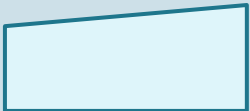
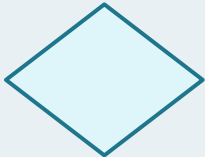
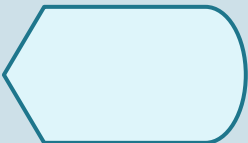
การเขียนผังงาน (Flow Chart)

- ❑ ผังงานเป็นเครื่องมือชนิดหนึ่งที่ใช้ในขั้นตอนการออกแบบโปรแกรม
- ❑ ผังงานแสดงให้เห็นขั้นตอนการทำงานของโปรแกรมหรือระบบใดๆ อย่างเป็นลำดับ
- ❑ ผังงานแสดงอยู่ในรูปของสัญลักษณ์ต่างๆ มีความหมายตามข้อกำหนดมาตรฐาน **ANSI : American National Standard Institute**





สัญลักษณ์ของผังงาน

สัญลักษณ์	ความหมาย
	Start/Stop ใช้แสดงจุดเริ่มต้นและจุดสิ้นสุดของขั้นตอนการแก้ปัญหา หรือโมดูลแต่ละโมดูล โดยจุดเริ่มต้นจะมีเส้นทางออกเพียงอย่างเดียว ส่วนจุดสิ้นสุดจะมีเส้นทางเพียงอย่างเดียว
	Process ใช้แสดงการประมวลผล การคำนวณ หรือการกำหนดค่าข้อมูลต่างๆ
	Input/Output ใช้แสดงการรับ หรือการแสดงข้อมูล หรือแสดงผลลัพธ์ โดยไม่ระบุชนิดของสื่อ

สัญลักษณ์ของผังงาน

สัญลักษณ์	ความหมาย
	Manual Input ใช้แสดงการรับข้อมูลโดยตรงจากแป้นพิมพ์
	Decision ใช้แสดงการเปรียบเทียบ หรือการตัดสินใจเลือกเส้นทางการทำงาน
	Display ใช้แสดงการแสดงผลลัพธ์ หรือแสดงข้อมูลทางจอภาพคอมพิวเตอร์

สัญลักษณ์ของผังงาน

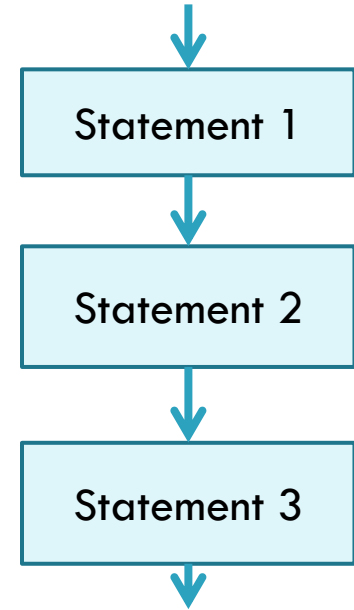
สัญลักษณ์	ความหมาย
	Document ใช้ในการแสดงผลลัพธ์ หรือแสดงข้อมูลส่งพิมพ์ลงบนกระดาษ
	On-Page Reference ใช้แสดงจุดเชื่อมต่อของผังงานที่อยู่ในหน้าเดียวกัน
	Off-Page Reference ใช้แสดงจุดเชื่อมต่อของผังงานที่อยู่คนละหน้า
	Flow Line ใช้แสดงทิศทางของลำดับขั้นตอนการทำงาน

รูปแบบการเขียนผังงาน

- มีด้วยกัน 3 รูปแบบ คือ
- 1. แบบเรียงลำดับ (Sequence Structure)
- 2. แบบมีเงื่อนไข (Decision Structure)
- 3. แบบวนซ้ำ หรือวนรอบ (Iteration Structure)

Sequence Structure

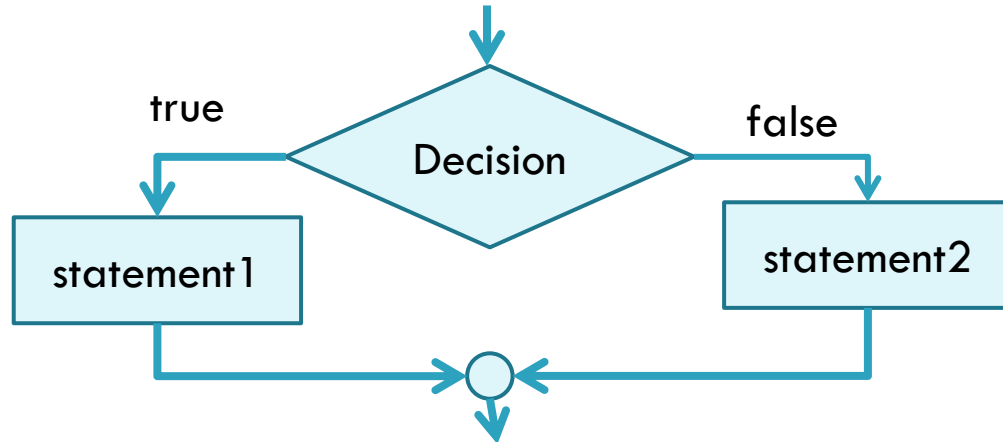
- ❑ มีลักษณะการทำงานแบบเรียงลำดับจากบนลงล่าง
- ❑ ไม่มีการทำงานแบบมีเงื่อนไข การตัดสินใจ
- ❑ ไม่มีการทำงานแบบวนซ้ำหรือวนรอบ



Decision Structure

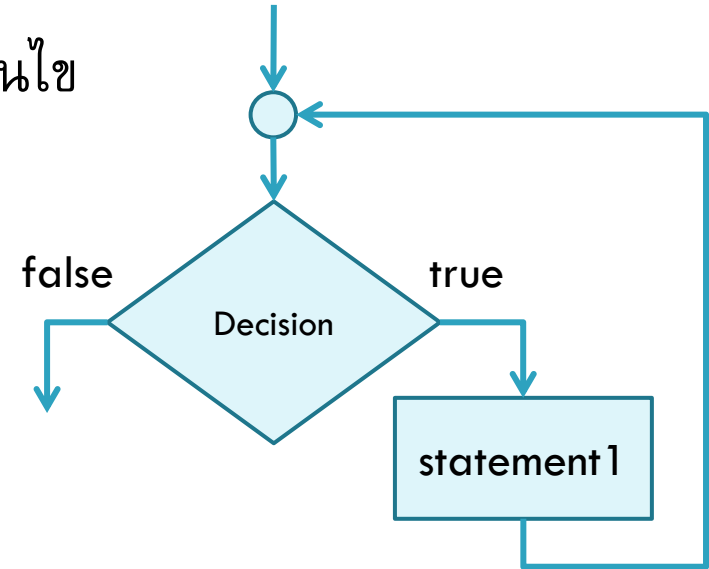
- ❑ มีลักษณะการทำงานแบบต้องมีการเปรียบเทียบข้อมูลตามเงื่อนไข
- ❑ เพื่อใช้สำหรับตัดสินใจเลือกดำเนินการตามขั้นตอนต่อไปหากเป็นจริง
- ❑ หรือหากเป็นเท็จก็จะดำเนินการตามขั้นตอนที่ได้กำหนดไว้ตามทิศทางของ

Flow line



Iteration Structure

- เป็นลักษณะการทำงานที่ต้องมีการทำซ้ำบ่อยๆ
- โดยจะนำงานนั้นๆ ไปใส่ไว้ใน **loop**
- ในการทำซ้ำแต่ละรอบจะมีการตรวจสอบเงื่อนไข
- ถ้าเงื่อนไขเป็นจริงก็จะยังคงทำงานใน **loop**
- แต่ถ้าเงื่อนไขเป็นเท็จก็จะออกจาก **loop**

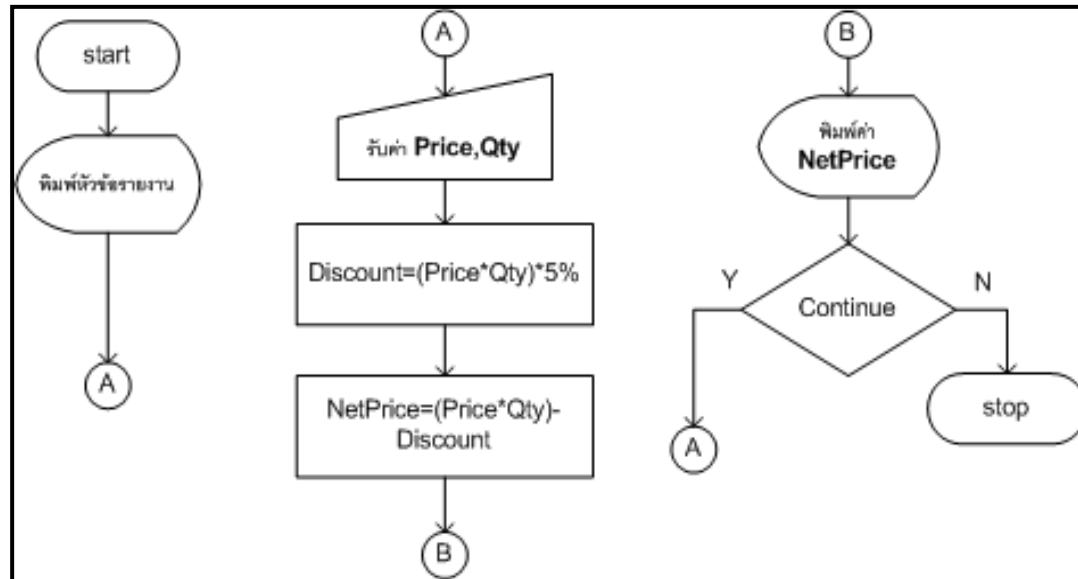


หลักการเขียนผังงาน

1. เขียนสัญลักษณ์ตามลำดับของ **Process** ที่ได้วิเคราะห์ไว้ หรืออาจใช้ผังงานแสดงให้เห็นขั้นตอนของอัลกอริทึมที่ได้ออกแบบไว้ก็ได้
2. ใช้สัญลักษณ์ให้ตรงกับความหมาย
3. ทุกๆ สัญลักษณ์จะต้องมีจุดเชื่อมต่อ จะอยู่ลอยๆ ไม่ได้
4. ทิศทางการเขียนผังงานจะต้องเริ่มจากข้างบนลงข้างล่าง
5. ใช้สัญลักษณ์การเชื่อมต่อผังงาน หากผังงานนั้นมีความซับซ้อน
6. การทำงานจะต้องมีจุดสิ้นสุดเสมอ
7. คำอธิบายการทำงานจะต้องกระชับ

ตัวอย่างการเขียนผังงาน

ตัวอย่างที่ 6 จาก process และอัลกอริทึมในตัวอย่างที่ 4 คำนวณหาราคาขายสุทธิ นำมาเขียนผังงานได้ ดังนี้



การเขียนซูโดโค้ด (Pseudocode)

- ❑ นอกจากการใช้ผังงานเพื่ออธิบายลำดับขั้นตอนของโปรแกรมใดๆ แล้ว
- ❑ เรายังสามารถใช้ ซูโดโค้ด (Pseudocode) อธิบายการทำงานของโปรแกรมคอมพิวเตอร์ได้เช่นกัน
- ❑ ซูโดโค้ด คือ คำอธิบายลำดับขั้นตอนของโปรแกรมคอมพิวเตอร์ โดยใช้คำภาษาอังกฤษที่มีโครงสร้างการเขียนตามแบบของคำสั่งในการเขียนโปรแกรม ดังนั้น ซูโดโค้ดจึงมีลักษณะคล้ายการเขียนชุดคำสั่งของโปรแกรม

รูปแบบการเขียนซูโดโค้ด

- เช่นเดียวกับผังงาน ที่ต้องมีรูปแบบเพื่อแสดงแทนโครงสร้างของโปรแกรมหลายรูปแบบ ได้แก่
- แบบเรียงลำดับ (Sequence Structure)
- แบบเปรียบเทียบหรือตัดสินใจ (Decision Structure)
- แบบวนซ้ำหรือวนรอบ (Iteration Structure)

Sequence Structure

□ ตัวอย่าง

```
ADD 1 to PageCount  
PRINT heading line1  
PRINT heading line2  
SET lineCount to zero  
READ customer record
```

Decision Structure

□ ตัวอย่าง

```
IF hoursWorked is greater than 40
    regularPay = hourlyRate*40
    overtimeHours = hoursWorked-40
    overtimePay = overtimeHours*hourlyRate*2
    grossPay = regularPay + overtimePay
ELSE
    grossPay = hoursWorked times hourlyRate
ENDIF
```

Iteration Structure

□ ตัวอย่าง

```
count = 0
WHILE count < 10
    READ name
    WRITE name
    ADD 1 to count
ENDWHILE
WRITE 'The End'
```

```
count = 0
REPEAT
    READ name
    WRITE name
    ADD 1 to count
UNTIL count >= 10
WRITE 'The End'
```

กฎเกณฑ์การเขียนซูโดโค้ด

1. เขียน 1 คำสั่งต่อ 1 บรรทัด
2. หากเป็นคำเฉพาะ (**Keyword**) ให้เขียนด้วยตัวพิมพ์ใหญ่ คำเฉพาะคือ คำสั่งให้ดำเนินการ ส่วนใหญ่จะเป็นคำแรกของประโยค และเป็นคำที่คล้ายกับคำสั่งในภาษาคอมพิวเตอร์ เช่น **ADD, READ, WRITE, IF** เป็นต้น
3. หากมีโครงสร้างการทำงานแบบ **Decision** และ **Iteration** คำสั่งในโครงสร้างควรมีการย่อหน้า
4. คำสั่ง **IF, WHILE...DO** และ **REPEAT...UNTIL** ควรมีคำสั่ง **END...** ปิดท้ายโครงสร้างเสมอ ช่วยให้อ่านซูโดโค้ดง่ายขึ้น

เอกสารอ้างอิง

พนิดา พาณิชกุล. การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้นด้วยภาษา **Java**.
กรุงเทพฯ : เค ที พี คอมพ์ แอนด์ คอนซัลท์, 2548.

End.

