

## การเขียนโปรแกรม

การทำงานของคอมพิวเตอร์

ระดับของภาษาในการ  
เขียนโปรแกรม

ตัวแปร

โครงสร้างภาษาสำหรับ  
การเขียนโปรแกรม

การออกแบบและเขียนโปรแกรม  
ที่มีการวนซ้ำด้วยคำสั่ง while

การออกแบบและเขียนโปรแกรม  
ที่มีการวนซ้ำด้วยคำสั่ง do while

# การทำงานของคอมพิวเตอร์



ใช้ภาษาระดับสูง

เขียนคำสั่งเพื่อสั่งงาน

ตัวแปลภาษา

แปลงคำสั่งเป็น  
ภาษาเครื่อง

คอมพิวเตอร์



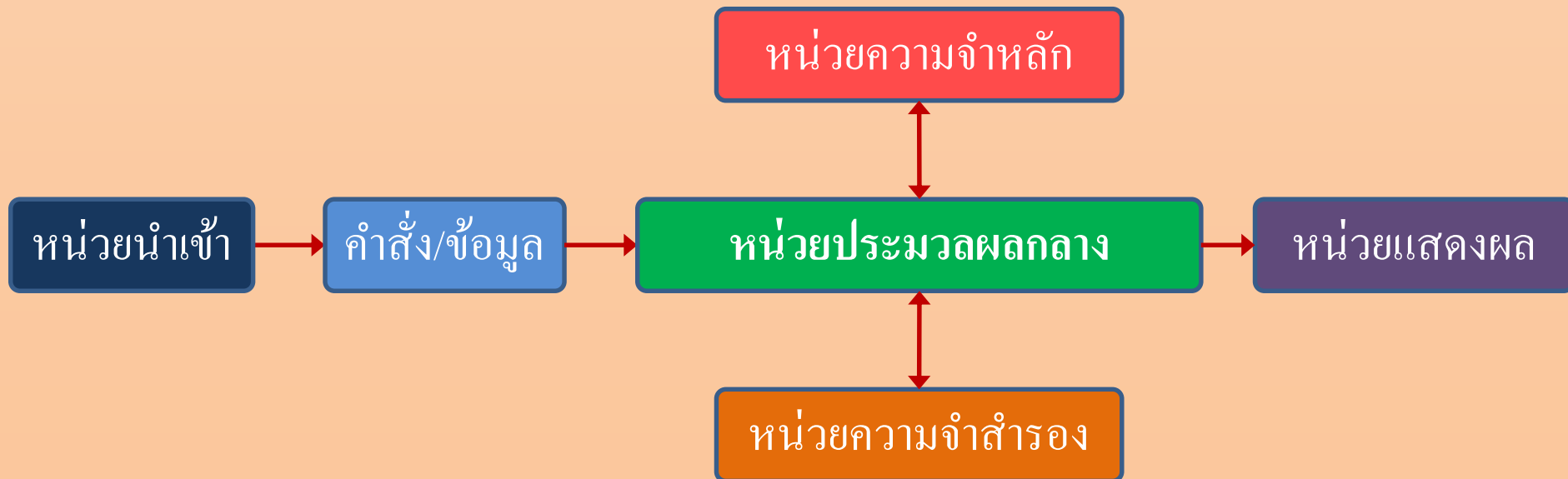
คอมพิวเตอร์เข้าใจ  
และปฏิบัติตามคำสั่ง

# การทำงานของคอมพิวเตอร์ประกอบด้วยองค์ประกอบพื้นฐาน 5 หน่วย ได้แก่

1. หน่วยนำเข้า (input unit) รับคำสั่งข้อมูลจากผู้ใช้คอมพิวเตอร์ ส่งต่อไปยังหน่วยประมวลผลกลาง

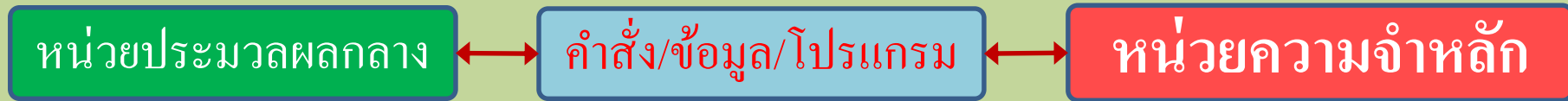


2. หน่วยประมวลผลกลาง (Central Processing Unit: CPU) ทำหน้าที่เปรียบเสมือนสมองของคอมพิวเตอร์ โดยจะคอยประมวลผลคำสั่งและทำงานประสานกับหน่วยอื่น ๆ

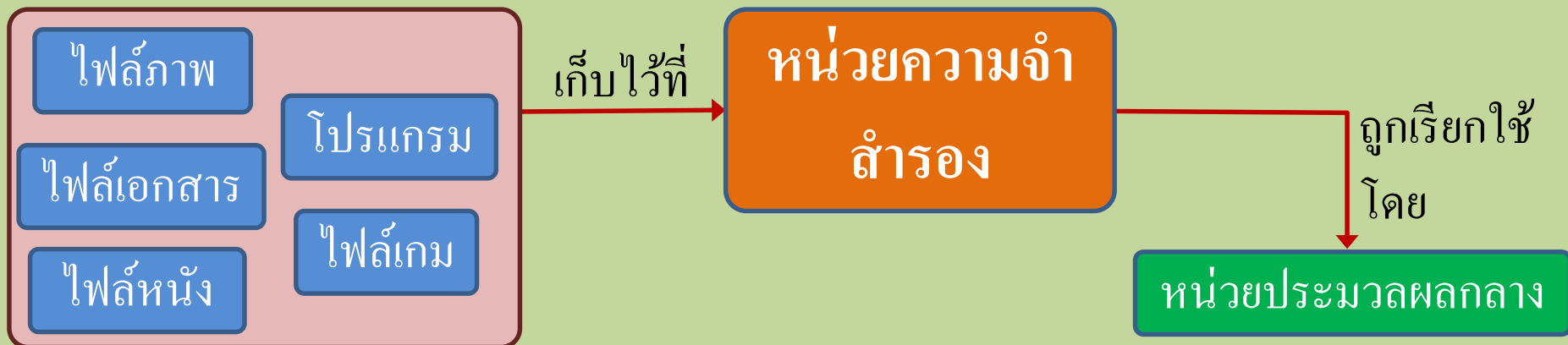


# การทำงานของคอมพิวเตอร์ประกอบด้วยองค์ประกอบพื้นฐาน 5 หน่วย

3. หน่วยความจำหลัก (main memory unit) ทำหน้าที่เก็บข้อมูลชุดคำสั่ง หรือ โปรแกรมต่าง ๆ ที่คอมพิวเตอร์กำลังประมวลผล และยังเป็นที่พักของข้อมูลระหว่างที่หน่วยประมวลผลกลาง กำลังประมวลผลอยู่



4. หน่วยความจำสำรอง (secondary memory unit) ทำหน้าที่เก็บข้อมูลหรือสำรองข้อมูลแบบถาวรบนคอมพิวเตอร์ อุปกรณ์ที่ใช้เก็บข้อมูลดังกล่าว เช่น ฮาร์ดดิสก์ ซีดีรอม แฟลชไดรฟ์



# การทำงานของคอมพิวเตอร์ประกอบด้วยองค์ประกอบพื้นฐาน 5 หน่วย

5. หน่วยแสดงผล (output unit) ทำหน้าที่แสดงผลหรือส่งออกข้อมูลที่ได้จากการประมวลผลของหน่วยประมวลผลกลางออกมาให้ผู้ใช้คอมพิวเตอร์ได้รับทราบ

หน่วยประมวลผลกลาง

หน่วยแสดงผล

ผู้ใช้คอมพิวเตอร์

สรุปภาพรวมการทำงาน

หน่วยความจำหลัก  
(main memory unit)



นำเข้า

(input unit)



หน่วยประมวลผลกลาง

(Central Processing Unit: CPU)

หน่วยแสดงผล  
(output unit)



หน่วยความจำสำรอง (secondary memory unit)



# ระดับของภาษาในการเขียนโปรแกรม

## 1. ภาษาเครื่อง (machine language)

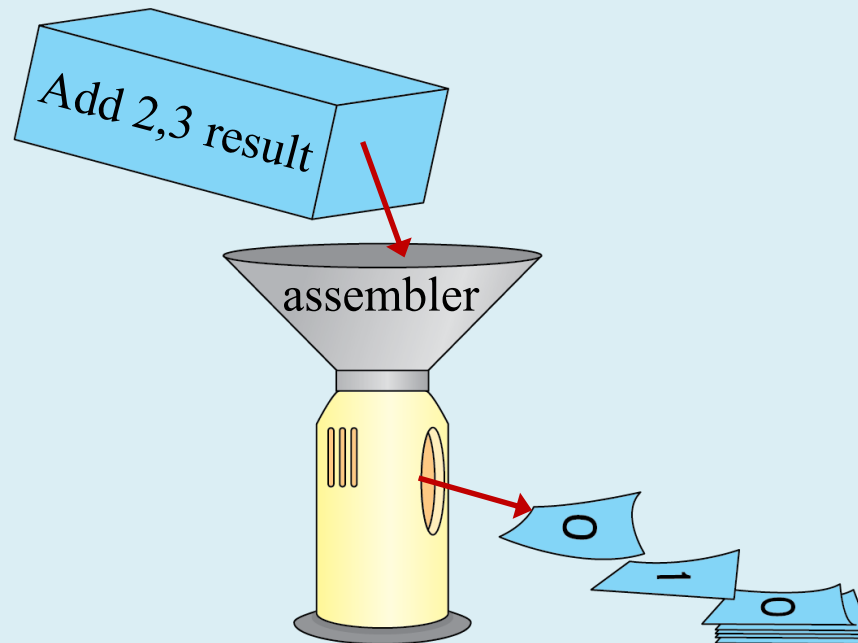
- ใช้ตัวเลขเพียง 2 ตัว ( binary code) คือ 0 และ 1 เรียกว่า เลขฐานสอง
- คอมพิวเตอร์สามารถเข้าใจและทำงานตามโคดโปรแกรมที่เขียน ไม่ต้องทำการแปลภาษา
- มีความซับซ้อนและยากต่อการจดจำ

```
00111010101111100010100000011100001001100010100100
0011111000101010001010111111000000000000010101010
1010001110100101010101010101010101001010101010101
010101010101110101010101000000111100101010100010101
010011111011000010101010001000001000100010101001010
100101001010000010101000101000010101000100010001001
001010000101010001101011111010100001100001111100000
001001000100011111000000001000000101100111110101010
011110000000001010101000010000001000000010110101010
101111010010101001010100101001000001111100001100001
000100101000001111110000010101000000111100101010100
0111111111100101010000010100011110010101001010000
111110001010000011100011100000010000101010000001001
000001111110000100001000100001000100010001010010101
0001010100010101010010101000101000110001010010101
010001010101000010010101010000101010001000100010000
101111100000001110000000000010101010000000101010000
010101111101001010000010010101000011100101010101010
1010101010101010100100100000010100010101000101010101
000100010101010010101001111000100000001010101010100
010101010100000011111000100000010100100101001010101
1010101010101010100010001010101010101011111001010
```

## 2. ภาษาแอสเซมบลี (assembly language)

- เป็นภาษาที่พัฒนาขึ้นมาเพื่อแก้ปัญหาจากความยุ่งยากของภาษาเครื่อง
- ใช้คำสั่งที่เป็นคำเฉพาะในภาษาอังกฤษที่มนุษย์สามารถเข้าใจได้มากขึ้น
- คอมพิวเตอร์ไม่สามารถทำงานด้วยภาษาแอสเซมบลีได้โดยตรงต้องใช้แอสเซมเบลอร์ (assembler) แปลภาษาแอสเซมบลีให้เป็นภาษาเครื่อง

### Assembly source file

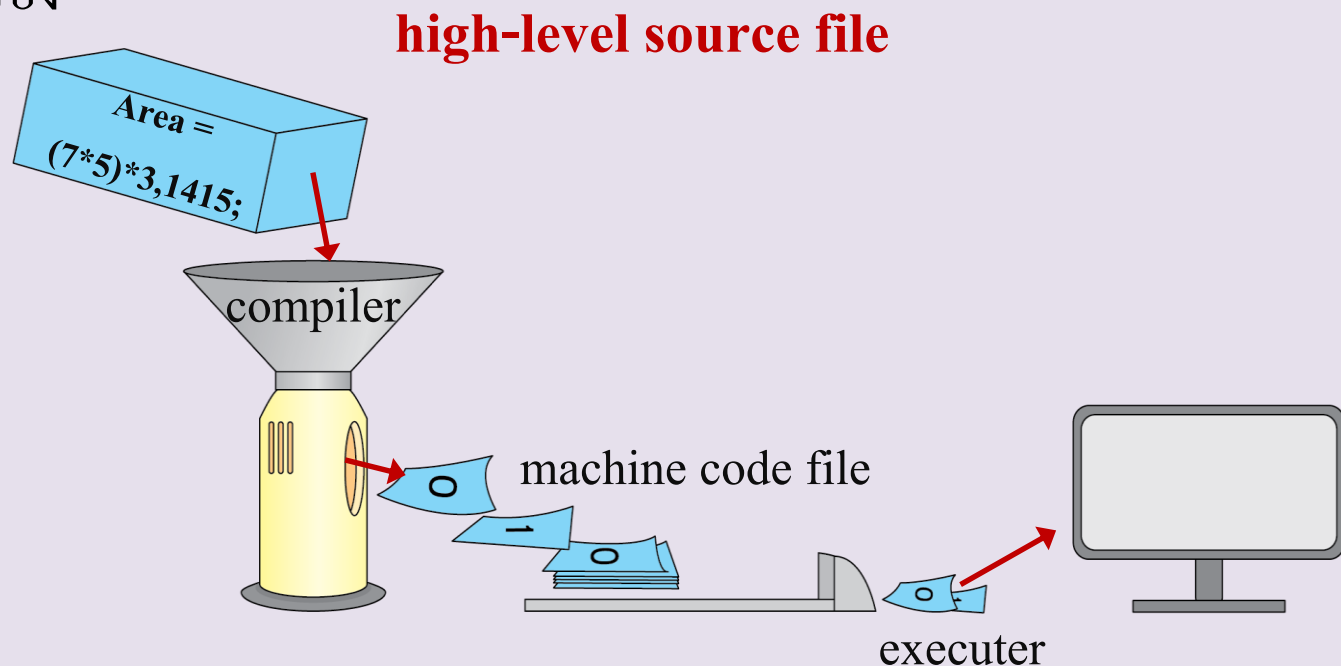


machine code file

แผนภาพ การแปลงคำสั่งจากแอสเซมบลีมาเป็นภาษาเครื่อง

### 3. ภาษาระดับสูง (high-level language)

- เป็นภาษาที่พัฒนาขึ้นมาเพื่อแก้ปัญหาจากความยุ่งยากของภาษาเครื่อง และภาษาแอสเซมบลี เพื่อให้มีความใกล้เคียงกับภาษามนุษย์มากขึ้น
- ใช้คำสั่งเป็นภาษาอังกฤษที่ง่ายต่อการสื่อสารและจดจำ
- คอมพิวเตอร์ไม่สามารถทำงานด้วยภาษาระดับสูงได้โดยตรงต้องมีการใช้คอมไพเลอร์ (compiler) หรืออินเทอร์พรีเตอร์ (interpreter) ในการแปลภาษาระดับสูงให้เป็นภาษาเครื่อง



แผนภาพ การแปลงคำสั่งจากภาษาระดับสูงมาเป็นภาษาเครื่อง



# ตัวแปร

ตัวแปร (variable) คือ สิ่งที่ใช้เก็บค่า (values) ที่สามารถเปลี่ยนค่าในโปรแกรมได้ในขณะประมวลผล ตัวแปรมีหลายชนิด แต่ละชนิดนั้นจะมีการจองหน่วยความจำ หรือพื้นที่ในการจัดเก็บที่ไม่เท่ากัน ทั้งนี้ขึ้นอยู่กับชนิดของข้อมูลที่เราต้องการจัดเก็บ



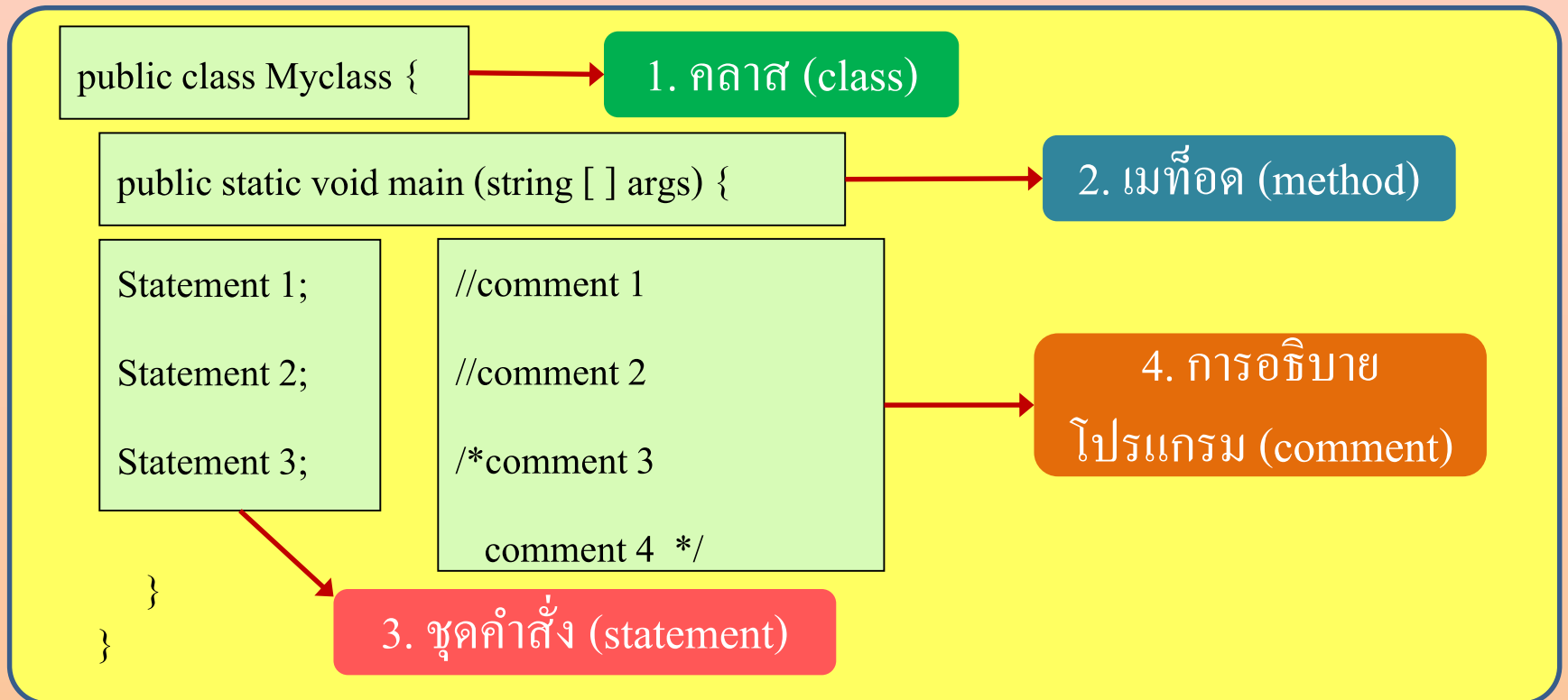
เปรียบเทียบภาชนะที่เหมาะสมและไม่เหมาะสมในการเก็บข้อมูล

เปรียบเทียบชนิดของข้อมูลกับหน่วยความจำที่ใช้เก็บข้อมูลที่เหมาะสม		
ชนิดของข้อมูล	ข้อกำหนด	ขนาด
boolean	ค่าใช่หรือไม่ใช่	1 บิต
byte	จำนวนเต็ม	8 บิต
short	จำนวนเต็ม	16 บิต
int	จำนวนเต็ม	32 บิต
long	จำนวนเต็ม	64 บิต
float	ทศนิยมไม่เกิน 6 ตำแหน่ง	32 บิต
double	ทศนิยมไม่เกิน 12 ตำแหน่ง	64 บิต
char	อักขระ	8 บิต

# โครงสร้างภาษาสำหรับการเขียนโปรแกรม

## 1. โครงสร้างของโคดภาษาจาวา (JAVA programming language)

คือ ภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง (James Gosling) และคณะ เพื่อใช้แทนภาษาซีพลัสพลัส (C++) การทำงานของภาษาจาวามีจุดเด่น คือ เมื่อเขียนแล้วสามารถนำไปใช้ได้กับอุปกรณ์ที่หลากหลาย เช่น สมาร์ทโฟน แท็บเล็ต เครื่องคอมพิวเตอร์ส่วนบุคคล โครงสร้างของโปรแกรมภาษาจาวาสามารถแสดงได้ ดังนี้



# โครงสร้างของโคตภาษาจาวา (JAVA programming language)

## 1. คลาส (class)

คลาส (class) ต้องมี 1 class  
เป็นอย่างน้อย

ต้องมีเครื่องหมาย  
ปีกกาคู่ { } กำกับ

การตั้งชื่อคลาสจะขึ้นต้น  
ด้วยตัวอักษรภาษาอังกฤษ  
ตัวพิมพ์ใหญ่

ชื่อของคลาสจะเป็น  
ชื่อเดียวกับชื่อ  
โปรแกรม

```
public class Myclass {  
}
```

# โครงสร้างของโคดภาษาจาวา (JAVA programming language)

## 2. เมทอด (method)

ใน 1 class จะต้องมี method  
อย่างน้อย 1 method

ภายใน method จะเขียนชุดคำสั่งซึ่ง  
เป็นการทำงานหลักของโปรแกรม

```
public class Myclass {  
    public static void main (string [] args) {  
    }  
}
```

### 3. ชุดคำสั่ง (Statement)

เป็นส่วนที่สั่งงานให้โปรแกรมทำงานตามที่ต้องการ

ทุกคำสั่งจะปิดท้ายคำสั่งด้วยเครื่องหมาย ; (Semicolon)

นิยมเขียน 1 คำสั่งต่อ 1 บรรทัด

### 4. การอธิบายโปรแกรม (comment)

เป็นส่วนที่ใช้ในการอธิบายเพิ่มเติม หรือจดบันทึกกลิ่นของผู้เขียนโปรแกรม ไม่มีผลใด ๆ ต่อการทำงานของโปรแกรม

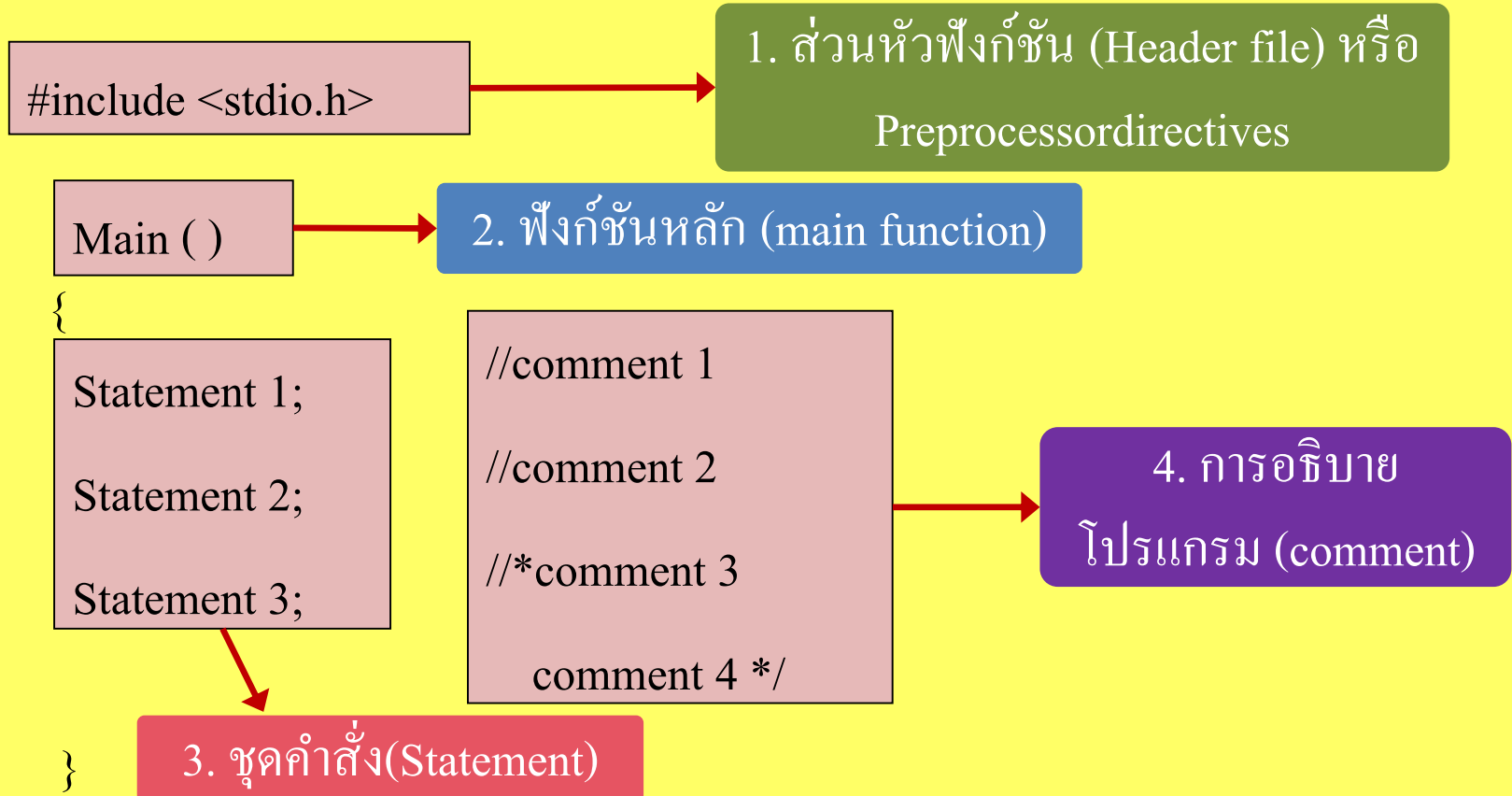
การอธิบายโปรแกรมทำได้ 2 ลักษณะ

(1) การอธิบายโปรแกรม 1 บรรทัด  
โดยการใส่เครื่องหมาย // และใส่คำอธิบายโปรแกรมหลังเครื่องหมาย

(2) การอธิบายโปรแกรมหลายบรรทัด  
โดยการใส่เครื่องหมาย /\* \*/ และจะใส่คำอธิบายโปรแกรมระหว่าง /\* กับ \*/

## 2. โครงสร้างของโคคภาษาซี (C programming language)

ภาษาซี หรือ C programming language มีลักษณะการเขียนโปรแกรมแบบโครงสร้าง พัฒนาโดย เดนนิส ริชชี (Dennis Ritchie) โดยมีจุดมุ่งหมายในการใช้เป็นภาษาสำหรับใช้ในการเขียนโปรแกรมปฏิบัติการระบบยูนิกซ์ ปัจจุบันภาษาซีเป็นภาษาหลักภาษาหนึ่งด้าน programming ที่นิยมใช้กันทั่วโลก โครงสร้างของโปรแกรมภาษาซีสามารถแสดงได้ ดังนี้



# โครงสร้างของโคคภาษาซี (C programming language)

## 1. หัวฟังก์ชัน (Header file) หรือ พรีโพรเซสเซอร์ไดเรกทีฟ (Preprocessor directives)

เป็นคำสั่งที่ใช้บอก compiler ว่าต้อง  
นำเข้าส่วนใดจาก library และจะติดต่อกับ  
อุปกรณ์นำเข้าและแสดงผลอะไรบ้าง

ใช้อุปกรณ์นำเข้าและแสดงผลพื้นฐาน  
คือ คีย์บอร์ดและจอภาพ โดยใช้ header ที่ชื่อ  
stdio.h และระบุตำแหน่ง directory ที่นำเข้า

การสั่งจะดำเนินการบอกตำแหน่ง directory  
ด้วยเครื่องหมาย # และใส่ชื่อ header ในเครื่องหมาย < > ดังนี้

#include <stdio.h> หมายถึง  
ไปค้นหา header file ชื่อ stdio.h  
ที่ตำแหน่ง directory include

#include <conio.h> หมายถึง  
ไปค้นหา header file ชื่อ conio.h  
ที่ตำแหน่ง directory include

# โครงสร้างของโคคภาษาซี (C programming language)

## 2. ฟังก์ชันหลัก (Main program)

เป็นส่วนที่ภาษาซี  
ทุกโปรแกรมจะต้องมี

ใช้ประกาศให้ตัวแปลคำสั่ง (compiler) รู้ว่าคำสั่ง  
ข้างในชุดปีกกา { } ของ main () เป็นชุดคำสั่ง  
ซึ่งเป็นการทำงานหลักของโปรแกรม

## 3. ชุดคำสั่ง (Statement)

เป็นส่วนที่สั่งงานให้โปรแกรม  
ทำงานตามที่เราต้องการ

ทุกคำสั่งจะปิดท้ายคำสั่งด้วย  
เครื่องหมาย ; (Semicolon)

นิยมเขียน 1 คำสั่งต่อ  
1 บรรทัด



# โครงสร้างของโคคภาษาซี (C programming language)

## 4. การอธิบายโปรแกรม (comment)

เป็นส่วนที่ใช้ในการอธิบายเพิ่มเติม หรือจดบันทึกสั้นๆของผู้เขียนโปรแกรม  
การอธิบายโปรแกรมจะไม่มีผลใด ๆ ต่อการทำงานของโปรแกรม

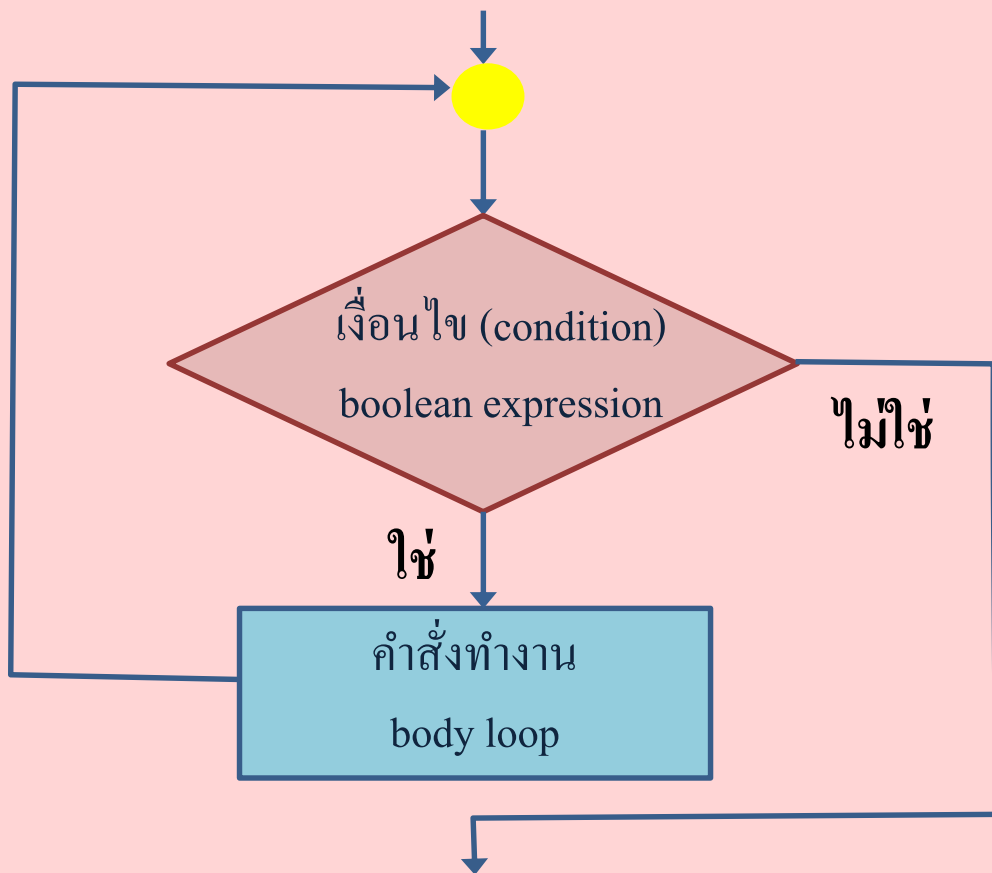
การอธิบายโปรแกรมทำได้ 2 ลักษณะ คือ

(1) การอธิบายโปรแกรม 1 บรรทัด  
สามารถทำได้โดยการใส่เครื่องหมาย //  
และใส่คำอธิบายโปรแกรม  
หลังเครื่องหมาย

(2) การอธิบายโปรแกรมหลายบรรทัด  
สามารถทำได้โดยการใส่เครื่องหมาย /\* \*/  
และจะใส่คำอธิบายโปรแกรมระหว่าง /\* กับ /\*

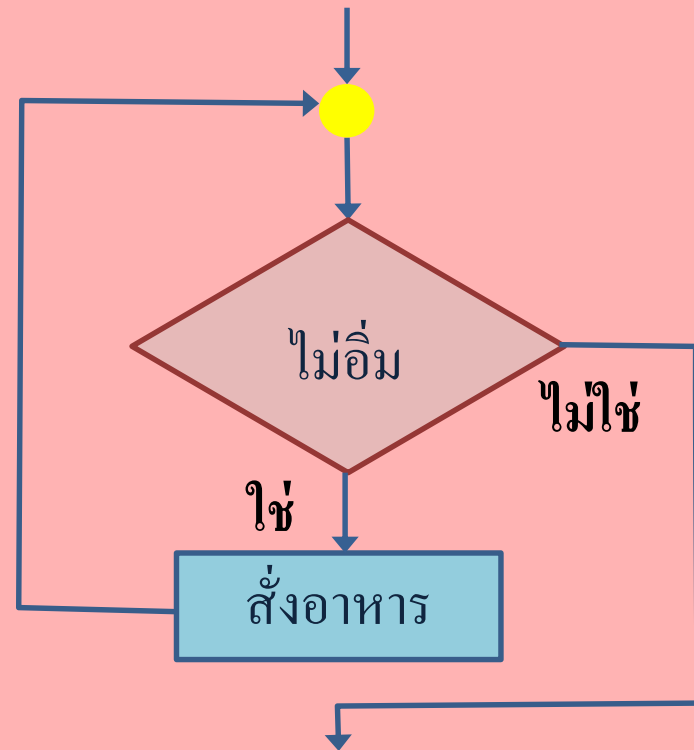
# การออกแบบและเขียนโปรแกรมที่มีการวนซ้ำด้วยคำสั่ง while

รูปแบบโปรแกรมวนซ้ำด้วยคำสั่ง while จะมีการตรวจสอบเงื่อนไขก่อนการทำงาน หากเหตุการณ์ยังอยู่ในเงื่อนไขก็จะทำงานที่กำหนดและวนซ้ำเพื่อทำงานจนกว่าเหตุการณ์จะไม่เป็นไปตามเงื่อนไขแล้วจึงออกจากการวนซ้ำ ซึ่งสามารถแสดงได้ดังผังงาน



# ตัวอย่างการออกแบบและเขียนโปรแกรมที่มีการวนซ้ำด้วยคำสั่ง while

เหตุการณ์ในชีวิตประจำวัน	โปรแกรมวนซ้ำ	เงื่อนไขทำงาน
เมื่อเราหิว เราสั่งอาหาร รับประทานในร้านและ ส่งไปเรื่อย ๆ จนกว่าจะอิ่ม เราจึงจะหยุดสั่งอาหาร	สั่งอาหาร	ไม่อิ่ม



# การออกแบบและเขียนโปรแกรมที่มีการวนซ้ำด้วยคำสั่ง do while

โปรแกรมวนซ้ำแบบมีเงื่อนไขด้วยคำสั่ง do while เป็นการทำงานก่อน แล้วจึงค่อยตรวจสอบเงื่อนไขการทำงานภายหลัง ในการเขียนโปรแกรมเราต้องการให้โปรแกรมทำงานก่อน 1 รอบ แล้วจึงค่อยมาตรวจเงื่อนไข

