

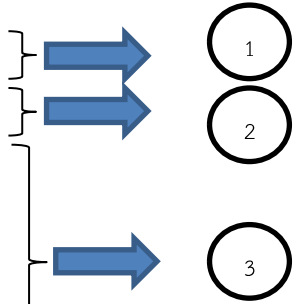
ใบความรู้ที่ 3

เรื่อง โครงสร้างโปรแกรมภาษาซี



โครงสร้างโปรแกรมภาษาซี

```
#include <stdio.h>
int main()
{
.....
getch();
return 0;
}
```



โครงสร้างของโปรแกรมภาษาซีแบ่งออกเป็น 3 ส่วน

1. ส่วนหัวของโปรแกรม

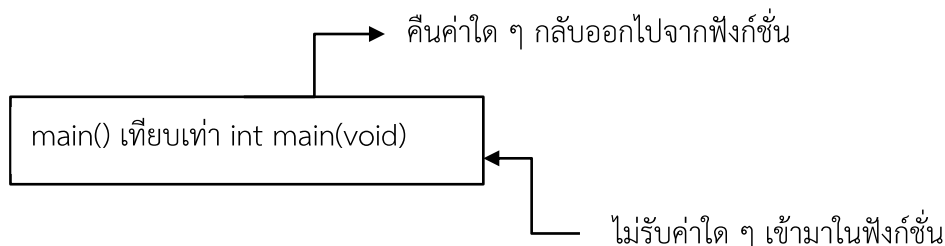
ส่วนหัวของโปรแกรมนี้นี้เรียกว่า Preprocessing Directive ใช้ระบุเพื่อบอกให้คอมไพเลอร์กระทำการใด ๆ ก่อนการแปลผลโปรแกรม ในที่นี้คำสั่ง #include <stdio.h> ใช้บอกกับคอมไพเลอร์ให้นำเฮดเดอร์ไฟล์ที่ระบุ คือ stdio.h เข้าร่วมในการแปลโปรแกรมด้วย โดยการกำหนด preprocessing directive จะต้องขึ้นต้นด้วยเครื่องหมาย # เสมอ

คำสั่งที่ใช้ระบุให้คอมไพเลอร์นำเฮดเดอร์ไฟล์เข้าร่วมในการแปลโปรแกรม สามารถเขียนได้ 2 รูปแบบคือ

- #include <ชื่อเฮดเดอร์ไฟล์> คอมไพเลอร์จะทำการค้นหาเฮดเดอร์ไฟล์ที่ระบุจากไถเรกทอรีสำหรับเก็บไคเฮดเดอร์ไฟล์โดยเฉพาะ (ปกติคือไคเรกทอรีชื่อ include)
- #include "ชื่อเฮดเดอร์ไฟล์" คอมไพเลอร์จะทำการค้นหาเฮดเดอร์ไฟล์ที่ระบุจากไคเรกทอรีเดียวกันกับไฟล์ source code นั้น แต่ถ้าไม่พบก็จะไปค้นหาไคเรกทอรีที่ใช้กับเฮดเดอร์ไฟล์โดยเฉพาะ

2. ส่วนของฟังก์ชันหลัก

ฟังก์ชันหลักของภาษาซีคือ ฟังก์ชัน main() ซึ่งโปรแกรมภาษาซีทุกโปรแกรมจะต้องมีฟังก์ชันนี้อยู่ในโปรแกรมเสมอ จะเห็นได้จากชื่อฟังก์ชัน คือ main แปลว่า "หลัก" ดังนั้นการเขียนโปรแกรมภาษาซีจึงขาดฟังก์ชันนี้ไปไม่ได้ โดยขอบเขตของฟังก์ชันจะถูกกำหนดด้วยเครื่องหมาย { และ } กล่าวคือ การทำงานของฟังก์ชันจะเริ่มต้นที่เครื่องหมาย { และจะสิ้นสุดที่เครื่องหมาย } ฟังก์ชัน main() สามารถเขียนอยู่ในรูปของ int main ก็ได้ คือหมายความว่า ฟังก์ชัน main() จะไม่มีอาร์กิวเมนต์ (argument) คือไม่มีการรับค่าใด ๆ เข้ามาประมวลผลภายในฟังก์ชัน และจะมีการคืนค่ากลับไปจากฟังก์ชันด้วย



3. ส่วนรายละเอียดของโปรแกรม

เป็นส่วนของการเขียนคำสั่งสั่ง เพื่อให้โปรแกรมทำงานตามที่ได้ออกแบบไว้

คอมเมนต์ในภาษาซี

คอมเมนต์ (comment) คือส่วนที่เป็นหมายเหตุของโปรแกรม มีไว้เพื่อให้ผู้เขียนโปรแกรมใส่ข้อความอธิบายกำกับลงไป ใน source code ซึ่งคอมไพเลอร์จะเข้ามาทำการแปลผลในส่วนที่เป็นคอมเมนต์นี้ คอมเมนต์ในภาษาซีมี 2 แบบคือ

- * คอมเมนต์บรรทัดเดียว ใช้เครื่องหมาย //
- * คอมเมนต์แบบหลายบรรทัด ใช้เครื่องหมาย /* และ*/

ตัวอย่าง การคอมเมนต์ในภาษาซี

```
// Comment only one line
#include <stdio.h>
#include <conio.h>
int main()
{
  clrscr();
  /* comment
   many
   line*/
  getch();
  return 0;
}
```

ข้อควรระวังในการใช้คอมเมนต์ คือ ในกรณีที่ใช้คอมเมนต์แบบหลายบรรทัด จะไม่สามารถใช้คอมเมนต์ซ้อนคอมเมนต์ได้ ดังรูป มิฉะนั้นจะก่อให้เกิดข้อผิดพลาดในการคอมไพล์

/*comment1*/	/*comment2*/	/*comment3*/	✓
/*comment1	/*comment2*/	/*comment3*/	✗

จะเห็นได้ว่าในกรณีที่ต้องการใส่คอมเมนต์หลาย ๆ บรรทัดติดกันนั้น คอมเมนต์แบบหลายบรรทัดจะ ช่วยประหยัดเวลาในการใส่คอมเมนต์ได้มากกว่าการใช้คอมเมนต์แบบบรรทัดเดียว แต่ก็ควรระมัดระวังในการใช้งานด้วย