

หน่วยที่ 2 ประวัติและโครงสร้างภาษาซี

ใบความรู้หน่วยที่ 2 ประวัติและโครงสร้างภาษาซี

ประวัติความเป็นมาภาษาซี

ภาษาซีเกิดขึ้นในปี ค.ศ. 1972 ผู้คิดค้นคือ Dennis Ritchie โดยพัฒนามาจากภาษาB และ ภาษา BCPL แต่ยังไม่มีการใช้งานอย่างกว้างขวางนัก ในปี ค.ศ. 1978 Brian Kernighan ได้ร่วมกับ Dennis Ritchie มาพัฒนามาตรฐานของภาษาซี เรียกว่า K&R ทำให้มีผู้สนใจเกี่ยวกับภาษาซีมากขึ้น จึงเกิดภาษาซีอีกหลายรูปแบบเพราะยังไม่มีข้อกำหนดรูปแบบภาษาซีที่เป็นมาตรฐาน และในปี 1988 Ritchie จึงได้กำหนดมาตรฐานของภาษาซีเรียกว่า ANSI C เพื่อใช้เป็นตัวกำหนดมาตรฐานในการสร้างภาษาซีรุ่นต่อไป ภาษาซี เป็นภาษาระดับกลางเหมาะสมสำหรับการเขียนโปรแกรมแบบโครงสร้าง เป็นภาษาที่มีความยืดหยุ่นมากคือใช้งานได้กับเครื่องต่างๆ ได้และปัจจุบันภาษาซีเป็นภาษาพื้นฐานของภาษาโปรแกรม

รุ่นใหม่ ๆ เช่น C++

ขั้นตอนการทำโปรแกรมด้วยภาษาซี

ขั้นตอนที่ 1 เขียนโปรแกรม (source code)

ใช้ editor เขียนโปรแกรมภาษาซีและทำการบันทึกไฟล์ให้มีนามสกุลเป็น .c เช่น work.c เป็นต้น editor คือ โปรแกรมที่ใช้สำหรับการเขียนโปรแกรม โดยตัวอย่างของ editor ที่นิยมนำมาใช้ในการเขียนโปรแกรมได้แก่ Notepad,Edit ของ Dos ,TextPad และ EditPlus เป็นต้น ผู้เขียนโปรแกรมสามารถเลือกใช้โปรแกรมใดในการเขียนโปรแกรมก็ได้ แล้วแต่ความถนัดของแต่ละบุคคล

ขั้นตอนที่ 2 คอมไพล์โปรแกรม (compile)

นำ source code จากขั้นตอนที่ 1 มาทำการคอมไพล์ เพื่อแปลจากภาษาซีที่มนุษย์เข้าใจไปเป็นภาษาเครื่องที่คอมพิวเตอร์เข้าใจได้ ในขั้นตอนนี้คอมไพเลอร์จะทำการตรวจสอบ source code ว่าเกิดข้อผิดพลาดหรือไม่

หากเกิดข้อผิดพลาด จะแจ้งให้ผู้เขียนโปรแกรมทราบ ผู้เขียนโปรแกรมจะต้องกลับไปแก้ไขโปรแกรม และทำการคอมไพล์โปรแกรมใหม่อีกครั้ง

หากไม่พบข้อผิดพลาด คอมไพเลอร์จะแปลไฟล์ source code จากภาษาซีไปเป็นภาษาเครื่อง (ไฟล์นามสกุล .obj) เช่นถ้าไฟล์ source code ชื่อ work.c ก็จะถูกแปลไปเป็นไฟล์ work.obj ซึ่งเก็บภาษาเครื่องไว้เป็นต้น

- **compile** เป็นตัวแปลภาษารูปแบบหนึ่ง มีหน้าที่หลักคือการแปลภาษาโปรแกรมที่มนุษย์เขียนขึ้นไปเป็น

- ภาษาเครื่อง โดยคอมไพเลอร์ของภาษาซี คือ C Compiler ซึ่งหลักการที่คอมไพเลอร์ใช้ เรียกว่า คอมไพล์ (compile) โดยจะทำการอ่านโปรแกรมภาษาซีทั้งหมดตั้งแต่ต้นจนจบ แล้วทำการแปลผลทีเดียว

นอกจากคอมไพเลอร์แล้ว ยังมีตัวแปลภาษาอีกรูปแบบหนึ่งเรียกว่า **อินเตอร์พรีเตอร์** การอ่านและแปลโปรแกรมทีละบรรทัด เมื่อแปลผลบรรทัดหนึ่งเสร็จก็จะทำงานตามคำสั่งในบรรทัดนั้น แล้วจึงทำการแปลผลตามคำสั่งในบรรทัดถัดไป หลักการที่อินเตอร์พรีเตอร์ใช้เรียกว่า **อินเตอร์พรีต (interpret)**

ข้อดีและข้อเสียของตัวแปลภาษาทั้งสองแบบมีดังนี้

	ข้อดี	ข้อเสีย
คอมไพเลอร์	<ul style="list-style-type: none"> - ทำงานได้เร็ว เนื่องจากทำการแปลผลทีเดียว แล้วจึงทำงานตามคำสั่งของโปรแกรมในภายหลัง-- - เมื่อทำการแปลผลแล้ว ในครั้งต่อไปไม่จำเป็นต้องทำการแปลผลใหม่อีก เนื่องจากภาษาเครื่องที่แปลได้จะถูกเก็บไว้ที่หน่วยความจำสามารถเรียกใช้งานได้ทันที 	<ul style="list-style-type: none"> - เมื่อเกิดข้อผิดพลาดขึ้นกับโปรแกรมจะตรวจสอบหาข้อผิดพลาดได้ยาก เพราะทำการแปลผลทีเดียวทั้งโปรแกรม
อินเตอร์พรีเตอร์	<ul style="list-style-type: none"> - หาข้อผิดพลาดของโปรแกรมได้ง่าย เนื่องจากทำการแปลผลทีละบรรทัด . เนื่องจากทำงานทีละบรรทัดดังนั้นจึงสั่งให้โปรแกรมทำงานตามคำสั่งเฉพาะจุดที่ต้องการได้ ไม่เสียเวลารอการแปลโปรแกรมเป็นเวลานาน 	<ul style="list-style-type: none"> - ช้า เนื่องจากที่ทำงานทีละบรรทัด

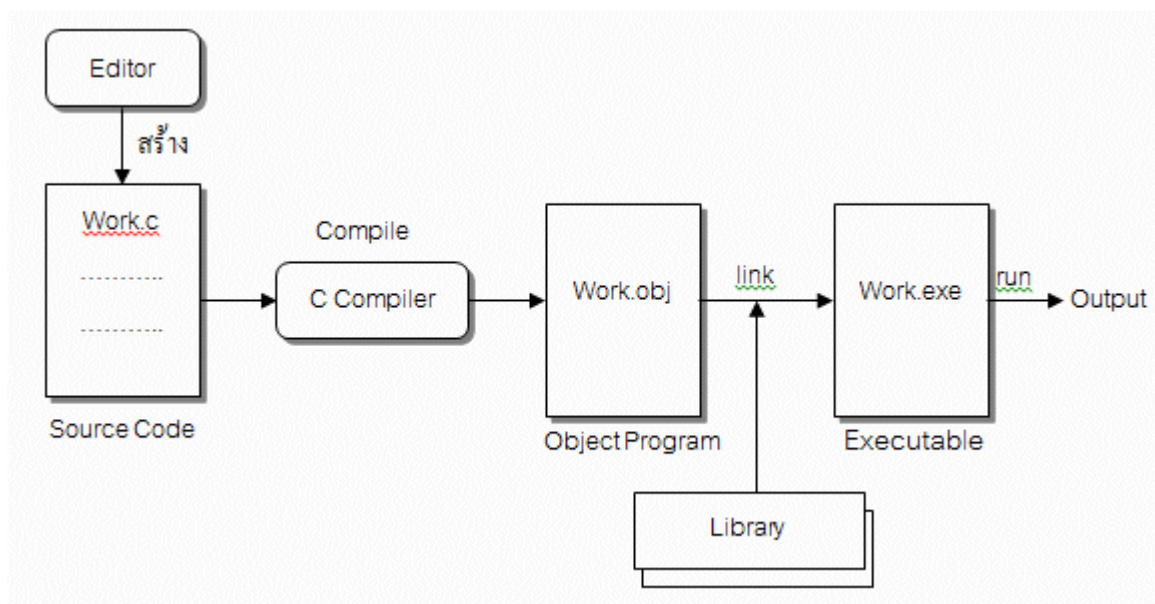
ขั้นตอนที่ 3 เชื่อมโยงโปรแกรม (link)

การเขียนโปรแกรมภาษาซีนั้นผู้เขียนโปรแกรมไม่จำเป็นต้องเขียนคำสั่งต่าง ๆ ขึ้นใช้งานเอง เนื่องจากภาษาซีมีฟังก์ชันมาตรฐานให้ผู้เขียนโปรแกรมสามารถเรียกใช้งานได้ เช่น การเขียนโปรแกรมแสดงข้อความ “Lumpangkanyanee” ออกทางหน้าจอ ผู้เขียนโปรแกรมสามารถเรียกใช้ฟังก์ชัน printf() ซึ่งเป็นฟังก์ชัน มาตรฐานของภาษาซีมาใช้งานได้ โดยส่วนการประกาศ (declaration) ของฟังก์ชันมาตรฐานต่าง ๆ จะถูกจัดเก็บอยู่ในเฮดเดอร์ไฟล์แต่ละตัว แตกต่างกันไปตามลักษณะการใช้งาน

ด้วยเหตุนี้ภาษาเครื่องที่ได้จากขั้นตอนที่ 2 จึงยังไม่สามารถนำไปใช้งานได้ แต่ต้องนำมาเชื่อมโยงเข้ากับ library ก่อน ซึ่งผลจากการเชื่อมโยงจะทำให้ได้ executable program (ไฟล์นามสกุล.exe เช่น work.exe) ที่สามารถนำไปใช้งานได้

ขั้นตอนที่ 4 ประมวลผล (run)

เมื่อนำ executable program จากขั้นตอนที่ 3 มาประมวลผลก็จะได้ผลลัพธ์ (output) ของโปรแกรมออกมา (ถ้ามี)



รูปที่ 1 ขั้นตอนการทำโปรแกรมด้วยภาษาซี

1. โครงสร้างของโปรแกรมภาษาซี

1.1 ข้อความสั่งตัวประมวลผลก่อน (preprocessor statements)

ข้อความสั่งตัวประมวลผลก่อนขึ้นต้นด้วยเครื่องหมาย # เช่น `#include<stdio.h>`

หมายความว่าให้ตัวประมวลผลก่อนไปอ่านข้อมูลจากแฟ้ม `stdio.h` ซึ่งเป็นแฟ้มที่มีอยู่ในคลังเมื่อโปรแกรมมีการใช้ข้อความสั่งอ่านและบันทึกข้อความสั่งตัวประมวลผลก่อนจะต้องเขียนไว้ตอนต้นขอโปรแกรม

1.2 รหัสต้นฉบับ (source code)

รหัสต้นฉบับหมายถึงตัวโปรแกรมที่ประกอบด้วยข้อความสั่งและตัวฟังก์ชันต่างๆ

1.3 ข้อความสั่งประกาศครอบคลุม (global declaration statements)

ข้อความสั่งประกาศครอบคลุมใช้ประกาศตัวแปรส่วนกลางโดยที่ตัวแปรส่วนกลางนั้นจะสามารถถูกเรียกใช้จากทุกส่วนของโปรแกรม

1.4 ต้นแบบฟังก์ชัน (function prototype)

ต้นแบบฟังก์ชันใช้ประกาศฟังก์ชันเพื่อบอกให้ตัวแปลโปรแกรมทราบถึงชนิดของค่าที่ส่งกลับและชนิดของค่าต่างๆที่ส่งไปกระทำในฟังก์ชัน

1.5 ฟังก์ชันหลัก (main function)

เมื่อสั่งให้กระทำการโปรแกรมฟังก์ชันหลักจะเป็นจุดเริ่มต้นของการกระทำการภายในฟังก์ชันหลักจะประกอบด้วยข้อความสั่งและข้อความสั่งที่เรียกใช้ฟังก์ชัน

1.6 ฟังก์ชัน (function)

ฟังก์ชันหมายถึงกลุ่มของข้อความสั่งที่ทำงานใดงานหนึ่งโดยเป็นอิสระจากฟังก์ชันหลักแต่อาจมีการรับส่งค่าระหว่างฟังก์ชันและฟังก์ชันหลัก

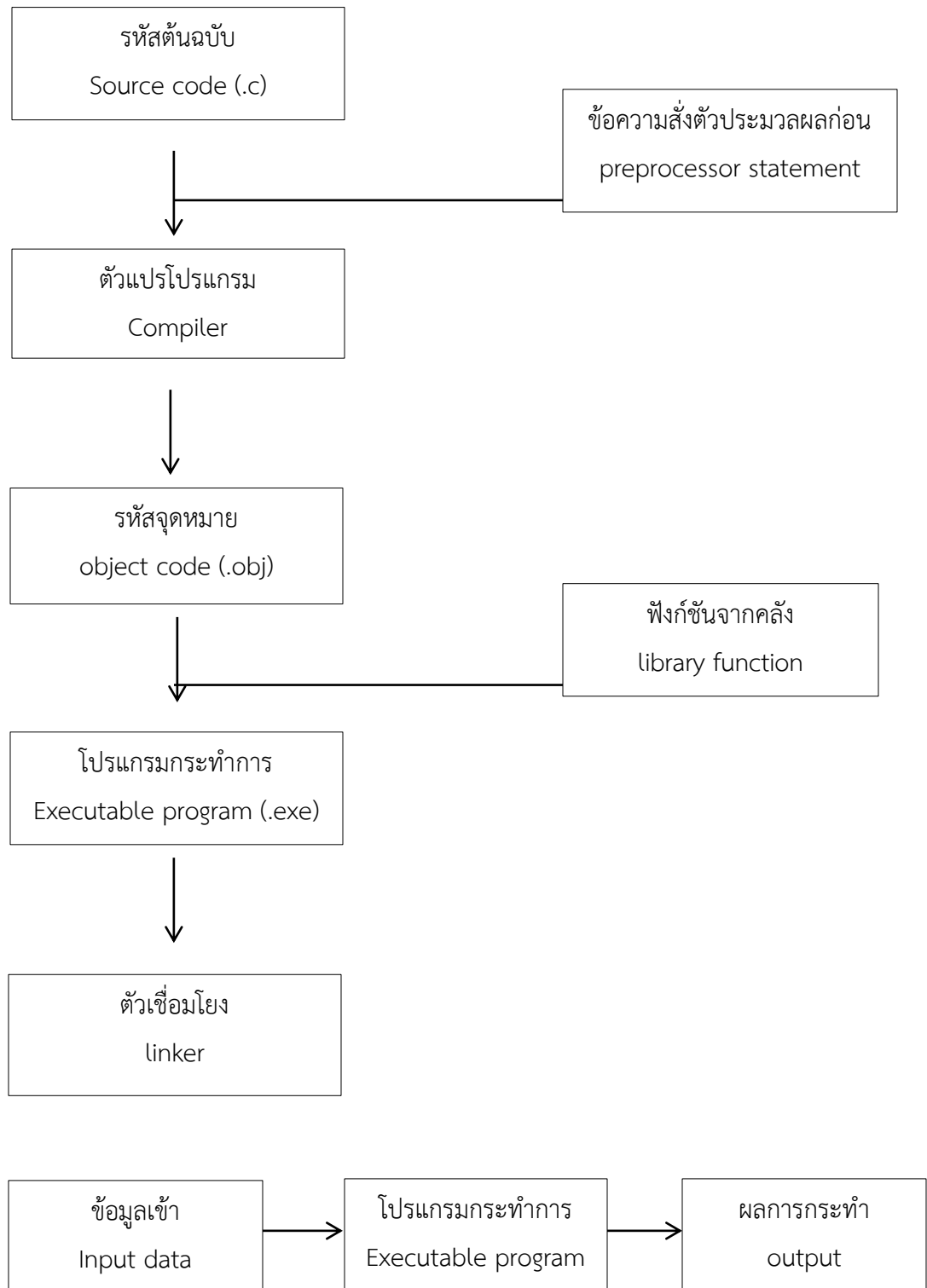
1.7 ข้อความสั่งประกาศตัวแปรเฉพาะที่ (local declaration statements)

ข้อความสั่งประกาศตัวแปรเฉพาะที่ใช้ประกาศตัวแปรเฉพาะที่โดยที่ตัวแปรเฉพาะที่จะสามารถถูกเรียกใช้เฉพาะภายในฟังก์ชันนั้น

1.8 การแปลและกระทำการโปรแกรม (program compilation and execution)

เมื่อได้เขียนและป้อนข้อความสั่งตัวประมวลผลก่อนและรหัสต้นฉบับลงในโปรแกรมอิดิเตอร์เสร็จแล้วจะต้องเรียกตัวแปลโปรแกรมมาเพื่อให้แปลภาษาซีให้เป็นภาษาเครื่องหากโปรแกรมนั้นเขียนได้ถูกต้องตรงตามกฎของภาษาซีตัวแปลโปรแกรมจะแปลโปรแกรมภาษาซีให้เป็นภาษาเครื่องแล้วนำไปเก็บไว้ในแฟ้มชื่อเดียวกันแต่มีนามสกุลเป็น `.obj` จากนั้นตัวเชื่อมโยง (linker) จะต้องนำฟังก์ชันจากคลัง (library function) ต่างๆที่โปรแกรมได้เรียกใช้มารวมเข้ากับแฟ้ม `.obj` แล้วนำไปเก็บไว้ในแฟ้มชื่อเดิมแต่มีนามสกุล

ไฟล์เป็น .exe เมื่อต้องการการทำงานของโปรแกรมก็สามารถป้อนข้อมูลเข้า (input data) ให้กับโปรแกรมซึ่งจะได้ผลการกระทำ (output)



รูปที่ 1 การแปลและการการทำงานของโปรแกรม

ตัวอย่างที่ 1.1

แสดงโครงสร้างของโปรแกรมโดยโปรแกรมนี้ประกอบด้วยฟังก์ชัน main() และฟังก์ชัน sum() ฟังก์ชัน main() ทาหน้าที่รับค่ามาเก็บไว้ในตัวแปร a และตัวแปร b แล้วส่งค่าของตัวแปรทั้งสองไปยังฟังก์ชัน sum() เพื่อคำนวณหาผลรวมเมื่คำนวณผลรวมแล้วจะส่งค่าของผลรวมกลับไปยังฟังก์ชัน main() จากนั้นฟังก์ชัน main() จะแสดงค่าของผลรวม

```
#include<stdio.h>           //คำสั่งตัวประมวลผลก่อน
int a, b, c;                //คำสั่งประกาศครอคลุม
int sum(int x, int y);      //คำสั่งแบบฟังก์ชัน
void main()                 //ฟังก์ชัน main()
{                             //เริ่มต้นฟังก์ชันmain()
scanf("%d", &a);            //คำสั่งรับค่าตัวแปร
scanf("%d", &b);            //คำสั่งรับค่าตัวแปร
c = sum(a, b);              //เรียกฟังก์ชัน sum()
printf("\n%d + %d = %d",a, b, c); //แสดงผล
}                             //จบฟังก์ชันmain()
int sum (int x, int y)      //ฟังก์ชัน sum()
{                             //เริ่มต้นฟังก์ชันsum()
return (x + y);             //คำสั่งรวมค่าและส่งค่ากลับ
}                             //จบฟังก์ชันsum()
```

2. ตัวแปร (variables)

คอมพิวเตอร์มีส่วนประกอบที่สำคัญส่วนหนึ่งคือหน่วยความจำซึ่งเปรียบได้กับสมองของมนุษย์ ทาหน้าที่เก็บข้อมูลในขณะที่ประมวลผลในการประมวลผลแต่ละครั้งมักต้องใช้ข้อมูลจำนวนมากซึ่งจำเป็นจะต้องเก็บไว้ในหน่วยความจำเป็นเก็บแล้วจะต้องทราบตำแหน่งที่นำข้อมูลเข้าไปเก็บไว้ภายในของหน่วยความจำด้วยเพื่อให้สามารถนำข้อมูลเหล่านั้นกลับมาประมวลผลได้ดังนั้นตัวแปรจึงมีหน้าที่สำคัญที่ช่วยในการเก็บข้อมูลแต่ละประเภทที่ผู้ใช้ป้อนเข้าสู่โปรแกรม

ชนิดข้อมูล (data types)

ข้อมูลที่ใช้ในโปรแกรมมีหลายชนิดซึ่งนักเขียนโปรแกรมต้องเลือกใช้ตามความเหมาะสมกับการใช้งานข้อมูลมีขนาดที่แตกต่างกันไปตามชนิดข้อมูลนอกจากนี้แล้วชนิดข้อมูลยังอาจมีขนาดที่แตกต่างกัน

โดยขึ้นกับเครื่องคอมพิวเตอร์และตัวแปลโปรแกรมที่ใช้ในการประมวลผลแต่โดยทั่วไปแล้วในไมโครคอมพิวเตอร์ชนิดข้อมูลมีการใช้ในโปรแกรมและขนาดดังนี้

ชนิดข้อมูล	การใช้ในโปรแกรม	คำสั่งในโปรแกรม	ขนาดข้อมูล (ไบต์)	ช่วงข้อมูล
character	char	%c	1	-128 ถึง 127
integer	int	%d	2	-32768 ถึง 32767
long integer	long	%ld	4	-2147483648 ถึง 2147483647
unsigned character	unsigned char	%c	1	0 ถึง 255
unsigned integer	unsigned int	%d	2	0 ถึง 65535
unsigned long	unsigned long	%ld	4	0 ถึง 4294967295
single-precision floating-point	Float	%f	4	1.2×10^{-38} ถึง 3.4×10^{38}

กฎการตั้งชื่อตัวแปร

การตั้งชื่อตัวแปรมีข้อกำหนดดังนี้

- ประกอบด้วย a ถึง z, 0 ถึง 9 และ _ เท่านั้น
- อักขระตัวแรกต้องเป็น a ถึง z และ _
- ห้ามใช้ชื่อเฉพาะ
- ตัวพิมพ์ใหญ่ตัวพิมพ์เล็กมีความหมายที่แตกต่างกัน
- มีความยาวได้สูงสุด 31 ตัวอักษร

การประกาศตัวแปร

การประกาศตัวแปรทำได้โดยเขียนข้อความสั่งขึ้นต้นด้วยชนิดข้อมูลตามด้วยชื่อตัวแปรและจบข้อความสั่งประกาศตัวแปรด้วยเครื่องหมายอัฒภาค (;) ดังนี้

ชนิดข้อมูลชื่อตัวแปร;

ถ้าต้องการประกาศตัวแปรชนิดเดียวกันหลายตัวต้องคั่นระหว่างตัวแปรด้วยเครื่องหมายจุลภาค (,) และจบด้วยเครื่องหมายอัฒภาค (;) ดังนี้

ชนิดข้อมูลชื่อตัวแปร1, ชื่อตัวแปร2;

เช่น

```
int count;           //ประกาศตัวแปรชื่อ count ใช้เก็บข้อมูลชนิด integer
int m, n;           //ประกาศตัวแปรชื่อ m และ n ใช้เก็บข้อมูลชนิด integer
intnum = 10;       //ประกาศตัวแปรชื่อ num และเก็บค่า 10 ไว้ในตัวแปรดังกล่าว
charstr = 'a';     //ประกาศตัวแปรชื่อ str และเก็บค่าอักขระ a ไว้ในตัวแปรดังกล่าว
```

3. การแสดงผลและการรับค่า

3.1 ฟังก์ชัน printf()

ฟังก์ชัน printf() เป็นฟังก์ชันจากคลังที่มาพร้อมกับตัวแปลโปรแกรมภาษาซี ใช้สำหรับการแสดงผลมีรูปแบบดังนี้

```
printf("สายอักขระควบคุม", ตัวแปร);
```

โดยที่สายอักขระควบคุมประกอบด้วย 3 ส่วนคือ

- ตัวอักขระที่จะแสดง
- รูปแบบการแสดงผลขึ้นต้นด้วยเครื่องหมายเปอร์เซ็นต์ (%)
- ลำดับหลัก (escape sequence)

ตัวแปรคือชื่อของตัวแปรที่จะแสดงผล

รูปแบบการแสดงผล (format specifiers)

การกำหนดรูปแบบการแสดงผล

- ขึ้นต้นด้วยเครื่องหมายเปอร์เซ็นต์ (%)
- ตามด้วยอักขระ 1 ตัวหรือหลายตัวโดยที่อักขระนั้นมีความหมายดังนี้

อักขระ	ชนิดข้อมูล	รูปแบบการแสดงผล
c	char	อักขระเดียว
d	int	จำนวนเต็มฐานสิบ
o		จำนวนเต็มฐานแปด
x		จำนวนเต็มฐานสิบหก
f	float	จำนวนที่มีทศนิยมใน รูปฐานสิบ

ลำดับหลัก (escape sequence)

ในการแสดงผลบางสิ่งบางอย่างที่จะแสดงอาจไม่ใช่ตัวอักษรจึงไม่สามารถที่จะเขียนสิ่งที่จะแสดงไว้ในโปรแกรมได้เช่นต้องการเขียนโปรแกรมให้ส่งเสียงหรือต้องการให้เลื่อนขึ้นบรรทัดใหม่ก่อน

แสดงข้อความดังนั้นในการเขียนโปรแกรมเพื่อแสดงผลสิ่งที่ไม่ใช่ตัวอักษรปกติจะต้องใช้ลำดับหลักเพื่อช่วยในการกำหนดอักขระพิเศษหรือสิ่งที่ไม่ใช่อักขระที่ต้องการให้โปรแกรมแสดง

ลำดับหลักจะเขียนขึ้นต้นด้วยเครื่องหมาย \ (back slash) แล้วตามด้วยอักขระในการทำงาน เครื่องหมายที่กลับหลังจะบอกให้เครื่องคอมพิวเตอร์ทราบว่าให้หลีกเลี่ยงการตีความอักขระที่ตามหลังมานี้ในลักษณะปกติเพราะอักขระเหล่านี้จะมีความหมายพิเศษแตกต่างออกไป

ลำดับหลัก	ผลการกระทำ
\n	ขึ้นบรรทัดใหม่
\t	เลื่อนไป 1 แท็บ
\a	เสียงกระดิ่ง
\b	ถอยไปหนึ่งวรรค
\f	ขึ้นหน้าใหม่
\\	แสดงเครื่องหมาย \
\'	แสดงเครื่องหมายฟันทอง
\"	แสดงเครื่องหมายฟันทู

3.2 ฟังก์ชันscanf()

ฟังก์ชันscanf() เป็นฟังก์ชันที่ใช้ในการรับข้อมูลจากแป้นพิมพ์โดยจะบอกเลขที่อยู่ของตัวแปรในหน่วยความจำแล้วจึงนำค่าที่รับมาเก็บไว้ตามที่อยู่นั้นโดยมีรูปแบบดังนี้

```
scanf(“%รูปแบบ”, &ตัวแปร);
```

โดยที่&ตัวแปรหมายถึงเลขที่อยู่ (address) ของตัวแปรที่จะรับค่ามาเก็บในหน่วยความจำ

4. การคำนวณทางคณิตศาสตร์

ในการเขียนโปรแกรมเพื่อทำการคำนวณทางคณิตศาสตร์จะต้องใช้ตัวดำเนินการต่างๆซึ่งมีวิธีการใช้งานและการทำงานดังนี้

การคำนวณ	ตัวดำเนินการ	ตัวอย่าง	การทำงาน
บวก	+	$c = a + b;$	นำค่าที่เก็บในตัวแปร a บวกกับค่าที่เก็บในตัวแปร b แล้วเก็บผลลัพธ์ไว้ใน c
ลบ	-	$c = a - b;$	นำค่าที่เก็บในตัวแปร a ลบด้วยค่าที่เก็บในตัวแปร b แล้วเก็บผลลัพธ์ไว้ใน c
คูณ	*	$c = a * b;$	นำค่าที่เก็บในตัวแปร a คูณกับค่าที่เก็บใน

			ตัวแปร b แล้วเก็บผลลัพธ์ไว้ใน c
หาร	/	$c = a / b;$	นำค่าที่เก็บในตัวแปร a หารด้วยค่าที่เก็บในตัวแปร b แล้วเก็บผลลัพธ์ไว้ใน c โดยถ้าทั้งตัวตั้งและตัวหารต่างเป็นจำนวนเต็มค่าที่เก็บในตัวแปร c จะเป็นจำนวนเต็ม แต่ถ้าตัวตั้งหรือตัวหารตัวใดตัวหนึ่งเป็นจำนวนจริงที่มีทศนิยมผลลัพธ์ที่ได้จะเป็นจำนวนจริงที่มีทศนิยมด้วย
มอดดูลัส	%	$c = a \% b;$	ให้ค่าที่เป็นในตัวแปร a เป็นตัวตั้งค่าที่เก็บในตัวแปร b เป็นตัวหารแล้วเก็บเศษจากการหารไว้ในตัวแปร c

5. ตัวดำเนินการเอกภาค (unary operator)

ตัวดำเนินการเอกภาคคือการใช้ตัวดำเนินการกับตัวแปรตัวเดียวในที่นี้จะแสดงการใช้ตัวดำเนินการ 2 ตัวกับตัวแปรตัวเดียวซึ่งมีลักษณะการใช้ 2 แบบคือ

การคำนวณ	ตัวดำเนินการ	ตัวอย่าง	การทำงาน
เพิ่มค่าตัวถูกดำเนินการทีละหนึ่ง	++	x++	$x = x + 1$
เพิ่มค่าตัวถูกดำเนินการทีละหนึ่ง	++	++x	$x = x + 1$
ลดค่าตัวถูกดำเนินการทีละหนึ่ง	--	x--	$x = x - 1$
ลดค่าตัวถูกดำเนินการทีละหนึ่ง	--	--x	$x = x - 1$

6. ตัวดำเนินการประกอบ (compound operator)

ตัวดำเนินการประกอบเป็นการใช้ตัวดำเนินการหนึ่งตัวร่วมกับเครื่องหมายเท่ากับกับการใช้ตัวดำเนินการประกอบจะช่วยให้เขียนข้อความสั้นได้สั้นและเร็วขึ้น


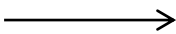

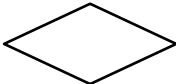
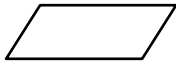

ตัวดำเนินการ	ตัวอย่าง	การทำงาน
+=	$x += 5$	$x = x + 5$
-=	$x -= 5$	$x = x - 5$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x \% = 5$	$x = x \% 5$

7. ผังงาน (flowchart)

ผังงานคือแผนภาพที่มีการใช้สัญลักษณ์รูปภาพและลูกศรที่แสดงถึงขั้นตอนการทำงานของโปรแกรมหรือระบบที่ละขั้นตอนรวมไปถึงทิศทางการไหลของข้อมูลตั้งแต่แรกจนได้ผลลัพธ์ตามที่ต้องการ
ประโยชน์ของผังงาน

- ช่วยลำดับขั้นตอนการทำงานของโปรแกรมและสามารถนำไปเขียนโปรแกรมได้โดยไม่สับสน
- ช่วยในการตรวจสอบและแก้ไขโปรแกรมได้ง่ายเมื่อเกิดข้อผิดพลาด
- ช่วยให้การดัดแปลงแก้ไขทำได้สะดวกและรวดเร็ว
- ช่วยให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมได้อย่างง่ายและรวดเร็วมากขึ้น

การเขียนผังโปรแกรมจะประกอบไปด้วยการใช้สัญลักษณ์มาตรฐานต่างๆที่เรียกว่าสัญลักษณ์ ANSI (American National Standards Institute) ในการสร้างผังงานดังตัวอย่างที่แสดงในรูปต่อไปนี้

สัญลักษณ์	ความหมาย
	จุดเริ่มต้น / สิ้นสุดของโปรแกรม
	ลูกศรแสดงทิศทางการทำงานของโปรแกรมและการไหลของข้อมูล
	ใช้แสดงคำสั่งในการประมวลผลหรือการกำหนดค่าข้อมูลให้กับตัวแปร
	การตรวจสอบเงื่อนไขเพื่อตัดสินใจโดยจะมีเส้นออกจากรูปเพื่อแสดงทิศทางการทำงานต่อไปเงื่อนไขเป็นจริงหรือเป็นเท็จ
	แสดงผลหรือรายงานที่ถูกสร้างออกมา
	แสดงจุดเชื่อมต่อของผังงานภายในหรือเป็นที่บรรจบของเส้นหลายเส้นที่มาจากหลายทิศทางเพื่อจะไปสู่การทำงานอย่างใดอย่างหนึ่งที่เหมือนกัน

ใบงานที่ 2.1
เรื่องความรู้พื้นฐานโปรแกรมภาษาซี

คำชี้แจง: ให้นักเรียนสรุปเนื้อหาและบันทึกความรู้ที่ได้จากการศึกษาเนื้อหาจากบทเรียนคอมพิวเตอร์ช่วยสอนบนระบบเครือข่ายเรื่องความรู้พื้นฐานโปรแกรมภาษาซีสรุปให้ได้ใจความสำคัญถูกต้องครบถ้วนตามหัวข้อที่กำหนดให้

1. บอกประวัติและกำเนิดภาษาซีมาพอสังเขป

.....

.....

.....

.....

.....

2. บอกข้อดีและข้อเสียของภาษาซีมาพอสังเขป

.....

.....

.....

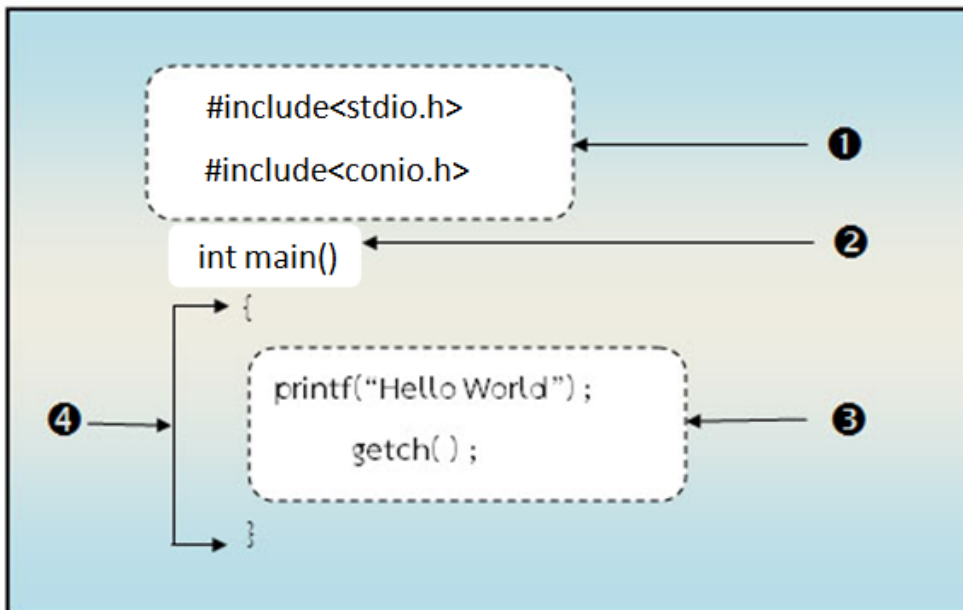
3. อธิบายขั้นตอนการคอมไพล์และลิงค์โปรแกรมในภาษาซีมาพอสังเขป

.....

.....

.....

4. จากภาพให้อธิบายโครงสร้างของภาษาซีตามหมายเลขมาพอสังเขป



4.1 หมายเลข 1 หมายถึง

.....

4.2 หมายเลข 2 หมายถึง

.....

4.3 หมายเลข 3 หมายถึง

.....

4.4 หมายเลข 4 หมายถึง

.....

5. การกำหนดตำแหน่งหมายเหตุ (Comment) ในภาษาซีมีกี่แบบ ?อะไรบ้าง ?

.....

.....

4. นักเรียนอธิบายความหมายของภาษาคอมไพเตอร์

.....

.....

5. อธิบายระดับของภาษาคอมไพเตอร์มี 5 ระดับ ได้แก่

.....

.....

6. จงยกตัวอย่างภาษาคอมไพเตอร์ 2 ภาษา พร้อมอธิบาย

6.1).....

6.2).....

7. จงกาเครื่องหมายถูก ในข้อที่ถูกต้อง และกาเครื่องหมายผิดในข้อที่ผิด
- 7.1 ภาษาคอมพิวเตอร์ที่เป็นภาษาระดับต่ำได้แก่ ภาษาเครื่อง (Machine Language)
 - 7.2 ตัวแปลภาษาของภาษา Assembly เราเรียกว่า Assembler
 - 7.3 ภาษาระดับสูงได้แก่ภาษา COBOL, BASIC, Pascal, C, C++
 - 7.4 ภาษาที่นำมาเขียนโปรแกรมประเภทระบบปฏิบัติการ ได้แก่ ภาษา C, C++
 - 7.5 ปัจจุบันนิยมใช้ภาษา BASIC เป็นเครื่องมือในการพัฒนาโปรแกรมในงานด้านธุรกิจ
 - 7.6..... ภาษาที่คอมพิวเตอร์เข้าใจได้ทันทีคือภาษาระดับต่ำ
 - 7.7 ภาษาระดับสูงเป็นภาษาที่ใกล้เคียงกับภาษามนุษย์ที่สุด
 - 7.8 อินเตอร์พรีเตอร์เป็นตัวแปลภาษาที่จะแปลภาษาที่ละบรรทัด
 - 7.9 ภาษาปาสคาลเป็นภาษาที่เหมาะสมกับการพัฒนาระบบปฏิบัติการ
 - 7.10 ภาษาจาวาเป็นภาษาที่สามารถทำงานได้กับเครื่องคอมพิวเตอร์ทุกระบบ

ใบงานที่ 2.2

รายวิชา ง 33202 การเขียนโปรแกรมภาษาซีเบื้องต้น ชั้นมัธยมศึกษาปีที่ 6

หน่วยการเรียนรู้ที่ 2 คอมพิวเตอร์และขั้นตอนการเขียนโปรแกรม

คำสั่ง นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

คอมพิวเตอร์ ตามความคิดของนักเรียนคืออะไร

.....

ให้เขียนแผนภาพแสดงขั้นตอนการทำงานของคอมพิวเตอร์ พร้อมอธิบาย

.....

จงเขียนชื่อแสดงฮาร์ดแวร์คอมพิวเตอร์ต่อไปนี้พร้อมบอกว่าเป็นอุปกรณ์ที่อยู่ในหน่วยใด



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อ

จัดอยู่ในหน่วย.....



ชื่อจัดอยู่ในหน่วย.....

ใบงานที่ 2.3

รายวิชา ง 33202 การเขียนโปรแกรมภาษาซีเบื้องต้น ชั้นมัธยมศึกษาปีที่ 6
หน่วยการเรียนรู้ที่ 1 ความรู้เบื้องต้นเกี่ยวกับคอมพิวเตอร์และขั้นตอนการเขียนโปรแกรม

คำสั่ง ให้นักเรียนตอบคำถามต่อไปนี้

ขั้นตอนการพัฒนาโปรแกรมคอมพิวเตอร์มีกี่ขั้นตอนอะไรบ้าง

.....

.....

.....

.....

.....

.....

.....

จงวิเคราะห์ปัญหาของการคำนวณการเปลี่ยนค่าอุณหภูมิจากหน่วยองศาเซลเซียส (Celsius) เป็นหน่วยองศาฟาเรนไฮต์(Fahrenheit) โดยสามารถคำนวณได้จากสูตร $F = (C \times 9) / 5 + 32$ โดยที่ F คืออุณหภูมิในหน่วยองศาฟาเรนไฮต์ , C คืออุณหภูมิในหน่วยองศาเซลเซียส

.....

.....

.....

.....

.....

.....

.....

.....

จงเขียนผังงานเพื่อแสดงขั้นตอนการเดินทางจากบ้านมาโรงเรียนของนักเรียน

จงเขียนผังงานแสดงขั้นตอนการคำนวณการเปลี่ยนค่าอุณหภูมิจากหน่วยองศาเซลเซียส (Celsius) เป็นหน่วยองศาฟาเรนไฮต์(Fahrenheit) จากข้อ 2 โดยให้รับข้อมูลค่าองศาเซลเซียสจากแป้นพิมพ์

จงเขียนผังงานแสดงขั้นตอนการคำนวณการเปลี่ยนค่าอุณหภูมิจากหน่วยองศาเซลเซียส (Celsius) เป็นหน่วยองศาฟาเรนไฮต์(Fahrenheit) จากข้อ 2 โดยให้รับข้อมูลค่าองศาเซลเซียสจากแป้นพิมพ์ โดยให้วนรอบรับข้อมูลจากแป้นพิมพ์ไปเรื่อย ๆ และจะออกจากการทำงานเมื่อผู้ใช้ป้อนค่าตัวเลขที่ต่ำกว่า 0