

ภาษาคอมพิวเตอร์ หมายถึง โปรแกรมหรือชุดคำสั่งที่โปรแกรมเมอร์เขียนเพื่อใช้สั่งงานตามรูปแบบและโครงสร้างของภาษาซึ่งแบ่งได้ 3 ระดับดังนี้คือ

1. ภาษาระดับต่ำ (Low Level Language) เป็นภาษาที่มนุษย์ทำความเข้าใจได้ยาก

ส่วนใหญ่ต้องมีความเข้าใจเกี่ยวกับสถาปัตยกรรมคอมพิวเตอร์ และฮาร์ดแวร์เป็นอย่างดีจึงจะสามารถเขียนโปรแกรมสั่งงานได้มีข้อดีในส่วนที่เขียนโปรแกรมควบคุมฮาร์ดแวร์แต่ละส่วนได้โดยตรงจึงทำงานได้เร็ว แต่ไม่เหมาะที่จะใช้ในการพัฒนาโปรแกรม ตัวอย่างของภาษาระดับต่ำได้แก่ ภาษาเครื่อง (Machine Language) และภาษาแอสเซมบลี (Assembly Language) เป็นต้น

2. ภาษาระดับกลาง (Medium Level Language) เป็นภาษาที่ทำความเข้าใจได้ไม่ยากนัก

เพราะมีลักษณะ เป็นภาษาแบบโครงสร้าง ทำความเข้าใจได้เหมือนกับภาษาระดับสูงแต่ทำงานได้รวดเร็วเหมือนกับภาษาระดับต่ำ สามารถใช้บนเครื่องที่มีความเร็วต่างกันโดยไม่ต้องดัดแปลง ภาษาระดับกลางจึงเป็นที่นิยมใช้กันแพร่หลาย ตัวอย่างของภาษาระดับกลาง ได้แก่ ภาษาซี เป็นต้น

3. ภาษาระดับสูง (High Level Language) เป็นภาษาที่ทำความเข้าใจได้ง่าย มีลักษณะของ

การใช้คำสั่งเป็นภาษาอังกฤษซึ่งใกล้เคียงกับภาษามนุษย์มากกว่าการสั่งให้คอมพิวเตอร์ทำงานจะต้องมีการแปลความหมายของคำสั่งโดยใช้ตัวแปลภาษาที่ละชุดคำสั่งที่เรียกว่า Interpreter หรือแปลครั้งเดียวทั้งโปรแกรมที่เรียกว่า Compiler

ภาษาซี (C Programming Language) ถูกพัฒนาขึ้นมาในปี ค.ศ. 1970 โดย Dennis Ritchie แห่ง Bell Laboratories และได้ถูกใช้งานแต่ในห้องปฏิบัติการของ Bell จนกระทั่งปี 1978 นั้น Brian Kernighan กับ Dennis Ritchie สองคู่หูหาใจ จึงได้ออกหนังสือ กำหนดมาตรฐานของภาษาซี ข้อกำหนดนี้คนมักเรียกขานกันว่า K&R C

หลังจากนั้นปี 1980 ภาษาซี ก็ได้รับความนิยมมากขึ้น มีการพัฒนา compiler ภาษาซีออกมามากมาย ความได้เปรียบของภาษาซี ที่เหนือกว่าภาษาอื่นคือ

1. ภาษาซี สามารถนำไปใช้ได้บนเครื่องทุก Platform ไม่ว่าจะเป็น Intel PC ที่วิ่ง Windows 95 หรือ Windows NT หรือ แม้แต่ Linux ทั้งเครื่อง Macintosh และ เครื่องเวอร์คสเตชัน ตลอดจนเมนเฟรม เนื่องจากมี compiler ของภาษาซี อยู่ทั่วไป
2. ภาษาซี เป็นภาษาที่ง่าย ๆ คือมีแต่ข้อกำหนดในการใช้งาน หรือ syntax แต่ไม่มีฟังก์ชันสำเร็จรูป (Built-in Function) ใดๆ ดังนั้นหากผู้ใช้ต้องการทำอะไรก็ตาม ต้องเขียนทุกอย่างขึ้นเอง หรือ อาจเรียก Library Functions มาใช้งาน โดย ฟังก์ชันที่เป็นงานที่ใช้บ่อยๆ จะถูกรวบรวมไว้ใน Library Functions เช่น การจัดการข้อความ การดำเนินการเกี่ยวกับ Input/Output (I/O) การจองหน่วยความจำ (Memory Allocation) แต่ฟังก์ชันที่วิไลสมารถหาจะไม่มีใน Standard Library เช่น ฟังก์ชันที่จัดการ Graphics ทั้งนี้ขึ้นอยู่กับระบบที่ใช้ (เช่น เป็นระบบ UNIX หรือ Windows 95) และ สิ่งแวดล้อมในการทำงาน (เช่น GUI เป็น X-Windows หรือ Direct X) การทำเช่นนี้จะทำให้ภาษาซี เป็นภาษาที่เคลื่อนย้ายได้ง่าย (portable)

เมื่อภาษาซี ได้รับความนิยมมากขึ้น จึงมีผู้ผลิต compiler ภาษาซีออกมาแข่งขันกันมากมาย ทำให้เริ่มมีการใส่ลูกเล่นต่างๆ เพื่อดึงดูดใจผู้ซื้อ ทาง American National Standard Institute (ANSI) จึงตั้งข้อกำหนดมาตรฐานของภาษาซีขึ้น เรียกว่า ANSI C เพื่อคงมาตรฐานของภาษาไว้ไม่ให้เปลี่ยนแปลงไป ใน lecture ของผมก็จะยึดแนวของ ANSI C นะครับ

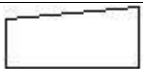
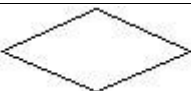
ในการเขียนโปรแกรมจะต้องเข้าใจหลักเกณฑ์ของภาษาโปรแกรม และระบบการทำงานของคอมพิวเตอร์ว่ามีโครงสร้างและวิธีการใช้คำสั่งอย่างไร หลักเกณฑ์การเขียน โปรแกรม ประกอบด้วย 5 ขั้นตอนคือ

- ✿ 1. ทำความเข้าใจและวิเคราะห์ปัญหา
- ✿ 2. กำหนดแผนในการแก้ปัญหา
- ✿ 3. เขียนโปรแกรมตามแผนที่กำหนด
- ✿ 4. ทดสอบและตรวจสอบความถูกต้อง
- ✿ 5. นำโปรแกรมที่ผ่านการทดสอบไปใช้งาน

ทำความเข้าใจและวิเคราะห์ปัญหา

ก่อนที่นักเรียนจะลงมือเขียนโปรแกรม นั้น นักเรียนจะต้อง ทำความเข้าใจและการวิเคราะห์ปัญหาเป็นสิ่งที่สำคัญ เพราะผู้เขียนโปรแกรมจะต้องวิเคราะห์ปัญหาร่วมกับนักวิเคราะห์ระบบว่า โจทย์ต้องการผลลัพธ์อะไร การที่จะได้มาซึ่งผลลัพธ์นั้น ต้องป้อนข้อมูลอะไรบ้าง และเมื่อป้อนข้อมูลเข้าไปแล้วจะทำการประมวลผล อย่างไร สิ่งเหล่านี้ผู้เขียนโปรแกรมจะต้องทำความเข้าใจให้ถูกต้อง เพราะถ้าผู้เขียนโปรแกรมวิเคราะห์ปัญหาไม่ถูกต้อง ผลลัพธ์ก็อาจจะไม่ตรงกับความต้องการของโจทย์ได้กำหนดแผนในการแก้ปัญหา

เมื่อทำความเข้าใจและวิเคราะห์ปัญหาโจทย์จนได้ข้อสรุปว่าโจทย์ต้องการอะไรแล้ว ก็ทำการกำหนดแผนในการแก้ปัญหาโดยการเขียนผังงาน (Flowchart) การเขียนผังงานคือ การเขียนแผนภาพที่เป็นลำดับ เพื่อแสดงขั้นตอนการทำงานของโปรแกรมเพื่อให้เข้าใจต่อการทำความเข้าใจ การเขียนผังงานมี 3 แบบคือ แบบเรียงลำดับ(Sequential) แบบมีการกำหนดเงื่อนไข(Condition) และแบบมีการทำงานวนรอบ(Looping) สัญลักษณ์ของผังงาน(Flowchart Symbol) มีดังนี้คือ

สัญลักษณ์	ความหมาย
	เริ่มต้นทำงาน
	กำหนดค่าหรือประมวลผล
	รับข้อมูลและแสดงผลข้อมูล
	รับข้อมูลทางแป้นพิมพ์
	การตัดสินใจ
	ใช้แสดงผลข้อมูลทางจอภาพ
	ใช้แสดงผลข้อมูลออกทางเอกสาร
	ทิศทางการดำเนินงาน
	ตัวเชื่อมต่อภายในหน้าเดียวกัน
	ตัวเชื่อมต่อไปหน้าอื่น

รูปแสดงสัญลักษณ์ที่ใช้ในการเขียนผังงาน

โครงสร้างของโปรแกรมภาษาซี และตัวอย่าง

โปรแกรมในภาษาซีทุกโปรแกรมจะประกอบด้วยฟังก์ชันอย่างน้อยหนึ่งฟังก์ชัน คือ ฟังก์ชัน main โดยโปรแกรมภาษาซีจะเริ่มทำงานที่ฟังก์ชัน main ก่อน ในแต่ละฟังก์ชันจะประกอบด้วย

1. Function Heading

ประกอบด้วยชื่อฟังก์ชัน และอาจมีรายการของ argument (บางคนเรียก parameter) อยู่ในวงเล็บ

2. Variable Declaration

ส่วนประกาศตัวแปร สำหรับภาษาซี ตัวแปรหรือค่าคงที่ทุกตัว ที่ใช้ในโปรแกรมจะต้องมีการประกาศก่อนว่าจะใช้งานอย่างไร จะเก็บค่าในรูปแบบใดเช่น interger หรือ real number

3. Compound Statements

ส่วนของประโยคคำสั่งต่างๆ ซึ่งแบ่งเป็นประโยคเชิงซ้อน (compound statement) กับ ประโยคนิพจน์ (expression statement) โดยประโยคเชิงซ้อนจะอยู่ภายในวงเล็บปีกกาคู่หนึ่ง { และ } โดยในหนึ่งประโยคเชิงซ้อน จะมีประโยคนิพจน์ที่แยกจากกันด้วยเครื่องหมาย semicolon (;) หลายๆ ประโยครวมกัน และ อาจมีวงเล็บปีกกาใส่ประโยคเชิงซ้อนย่อยเข้าไปอีกได้ ซึ่งนักศึกษาจะได้เห็นต่อไป

ข้างล่างนี้คือตัวอย่างโปรแกรมง่ายๆ ในภาษาซี

```
โปรแกรมที่ 1: Hello
#include <stdio.h>

main()
{
    printf(" Hello. This is my first program. \n");    /* This is a comment */
    return 0 ;
}
```

บรรทัดแรกนั้นเราเรียก Compiler Directives คือเป็นคำสั่งที่บอก compiler ว่ามีไฟล์อะไรที่จำเป็นต่อการ compile บ้าง ซึ่งในที่นี้ เราต้องการไฟล์ที่ชื่อ "stdio.h" ซึ่งทำหน้าที่เรียกใช้งาน Standard I/O Library ซึ่งฟังก์ชันที่เราเรียกใช้งานใน

โปรแกรมข้างบนคือ **printf**

บรรทัดต่อมาบอกว่าโปรแกรมนี้มีฟังก์ชัน main โดยไม่ต้องการ argument ใดๆ โดย compound statement ถูกบรรจุในวงเล็บปีกกา { }

บรรทัดต่อมามีการเรียกใช้ฟังก์ชัน **printf** จาก Standard I/O Library โดย argument ของฟังก์ชันนี้ก็คือประโยค " Hello. This is my first program." นักศึกษาสังเกตว่ามีชุดอักขระ \n ซึ่งเป็นชุดอักขระพิเศษหมายถึงการขึ้นบรรทัดใหม่ นี่คือนักศึกษาสังเกตว่า และ ความหมาย

\a	bell
\b	back space
\t	horizontal tab
\v	vertical tab
\n	new line
\f	form feed
\r	carriage return
\"	quotation mark (")
\'	apostrophe (')
\?	question mark (?)
\\	back slash (\)
\0	null

สามารถใช้ comment ในโปรแกรมเพื่ออธิบายสิ่งที่ทำได้ โดย comment จะต้องอยู่ระหว่าง /* และ */ เสมอ
บรรทัดเกือบสุดท้ายมีคำสั่ง return 0 ซึ่งบอกว่าฟังก์ชันมีการส่งค่ากลับ โดยค่า 0 ที่ส่งกลับไปบอกว่าฟังก์ชัน main ได้ทำงานสมบูรณ์แล้ว

โปรแกรมที่ 2

```
#include <stdio.h>

main ()
{
    int      width , length ;          /* Declaration of Variables */
    int      area ;

    width = 10 ;
    length = 5 ;                        /* Expression Statements */
    area = width * length ;

    printf ("A rectangular having width = %d and length = %d has area = %d \n", width, length, area) ;
    return 0
}
```

สิ่งที่เพิ่มขึ้นมาในโปรแกรมนี้นี้คือ มีการใช้ตัวแปร จึงต้องมีการประกาศตัวแปรด้วย ซึ่งตัวแปรทั้งหมด เราใช้แบบเลขจำนวนเต็ม หรือ แบบ integer (จะกล่าวถึงหลักการในการตั้งชื่อต่อไป) นอกจากนั้นในฟังก์ชัน printf ยังมี argument เพิ่มขึ้น คือ มีการพิมพ์ค่าของตัวแปร width length และ area ด้วย เราจึงต้องระบุ Format ของการพิมพ์ตัวแปรดังกล่าว ในที่นี้เราระบุให้เป็นเลขฐานสิบ หรือ Decimal เราจึงใช้ %d ในการระบุ Format ของสิ่งที่จะพิมพ์ออกมา

โปรแกรมที่ 3

```
#include <stdio.h>

main()
{
    float    width , length ;
    float    area ;

    printf("Please enter width: ") ;
    scanf("%f", &width) ;
    printf("Please enter length: ") ;
    scanf("%f", &length) ;

    area = width * length ;

    printf ("A rectangular having width = %d and length = %d has area = %d \n", width, length, area) ;
    return 0
}
```

ในโปรแกรมที่ 3 เราเปลี่ยนการใช้ตัวแปรจาก integer ไปเป็นแบบ floating point เพื่อให้ตัวแปรสามารถบรรจุ เลขทศนิยมได้ โปรแกรมนี้มีความสามารถมากขึ้น คือสามารถรับข้อมูลเข้าจากผู้ใช้ได้ด้วยการใช้ฟังก์ชัน scanf โดยต้องกำหนด Format ของสิ่งที่ได้รับใน argument โดยใส่ในเครื่องหมายคำพูด และ ตัวแปรที่จะใช้เก็บค่าต้องขึ้นต้นด้วยเครื่องหมาย &

การกำหนด Format สำหรับข้อมูล เราต้องระมัดระวัง ต้องกำหนดให้ตรงกับประเภทของข้อมูลด้วย %d นั้นใช้สำหรับ เลขจำนวนเต็ม และ %f ใช้สำหรับ floating point จะเกิดอะไรขึ้นถ้านักศึกษาจะเลขจุดนี้ไป

โปรแกรมที่ 4 การกำหนด Format ข้อมูลผิดพลาด

```
#include <stdio.h>

main()
{
    int      i ;

    printf("The number is %f\n",i) ;    /* Using %f with integer */
    return 0
}
```

การตั้งชื่อตัวแปร และ ฟังก์ชัน

หลักการในการตั้งชื่อตัวแปรและฟังก์ชันมีดังต่อไปนี้

1. สามารถใช้ตัวอักษร A ถึง Z รวมทั้ง ตัวเลข 0 ถึง 9 และ ขีดล่าง (_) ด้วย โดยมีข้อแม้ว่าต้องไม่ใช่ตัวเลขนำหน้าชื่อตัวแปร เช่น 2noy เป็นตัวแปรที่ไม่ถูกต้อง แต่ noy2 ใช้ได้
2. สำหรับ ANSI C ชื่อตัวแปรมีความยาวได้ถึง 31 ตัวอักษร
3. ชื่อตัวแปรและฟังก์ชัน ไม่อาจใช้คำสงวน (Reserved Words) ต่อไปนี้

4.	auto	extern	sizeof
5.	break	float	static
6.	case	for	struct
7.	char	goto	switch
8.	const	if	typedef
9.	continue	int	union
10.	default	long	unsigned
11.	do	register	void
12.	double	return	volatile
13.	else	short	while
14.	enum	signed	

ประเภทของข้อมูล (Data Types)

ภาษาซีแบ่งการใช้ข้อมูลออกเป็นประเภทต่างๆ เพื่อให้เหมาะสมกับการใช้งาน โดยข้อมูลแต่ละประเภทก็จะมีขนาดพื้นที่เก็บต่างกันด้วย ตลอดจนช่วงของข้อมูลที่ใช้ก็จะแตกต่างกัน

Type	Data Range	Size
int	-32,768 to 32,767	2 bytes
long int	-2,147,483,648 to 2,147,483,647	4 bytes
float	8 digit floating point	4 bytes
double	16 digit floating point	8 bytes
char	character	1 byte

หากเรากำหนดตัวแปรโดยเพิ่มคำว่า **unsigned** เข้าไป เช่น unsigned int จะทำให้ตัวแปรใช้เฉพาะส่วนที่เป็นค่าบวก ทำให้สามารถมี data range ส่วนที่เป็นบวกเพิ่มขึ้นเป็น 2 เท่าได้ เช่น unsigned int มี data range จาก 0 ถึง 65,535