หน่วยการเรียนรู้ที่ 3 สร้างสรรค์งานด้วยภาษาซี
ใบความรู้ที่ 4 เรื่อง เริ่มต้นกับภาษาซี

การแก้ปัญหาด้วยคอมพิวเตอร์นั้น หลังจากที่ได้วิเคราะห์ปัญหาจนได้ขั้นตอนวิธีในการแก้ปัญหา ซึ่งอาจอยู่ในรูปแบบรหัสลำลอง หรือผังงาน ขั้นตอนต่อไปคือการเขียนโปรแกรมคอมพิวเตอร์ขึ้น เพื่อแก้ปัญหาตามขั้นตอนที่ได้วางแผนไว้ แต่เนื่องจากคอมพิวเตอร์จะรับรู้คำสั่งที่เป็นภาษาเครื่องเท่านั้น และมนุษย์ไม่สามารถเขียนโปรแกรมภาษาเครื่องได้โดยตรง เนื่องจากไม่สะดวก ยากต่อการทำความเข้าใจ จึงได้มีการสร้างภาษาคอมพิวเตอร์ระดับสูงขึ้นมาเพื่อให้ง่ายต่อการเขียนโปรแกรม ผู้เขียนโปรแกรมไม่จำเป็นต้องเรียนรู้การทำงาน และโครงสร้างภายในของเครื่องคอมพิวเตอร์ ก็สามารถเขียนโปรแกรมได้ โดยต้องผ่านการแปลให้เป็นภาษาเครื่องก่อนที่จะใช้สั่งงานคอมพิวเตอร์ได้

Tip & Trick

ภาษาคอมพิวเตอร์แบ่งได้ 3 ประเภท คือ ภาษาเครื่อง ภาษาระดับต่ำ และภาษาระดับสูง

ภาษาเครื่อง คือ ภาษาที่มีคำสั่งอยู่ในรูปเลขฐานสอง **ภาษาระดับต่ำ** คือ ภาษาที่ใกล้เคียงกับภาษาเครื่อง โดยใช้ตัวอักษรภาษาอังกฤษแทนรหัสแสดงการทำงาน และใช้การตั้งชื่อตัวแปรแทนตำแหน่งที่ใช้เก็บค่าต่าง ๆ แต่เนื่องจากคอมพิวเตอร์สามารถเข้าใจคำสั่งในรูปแบบเลขฐานสองได้เพียงอย่างเดียว ดังนั้นถ้าเขียนโปรแกรมให้เป็นภาษาระดับต่ำ เมื่อแปลภาษาเครื่องจะทำให้คอมพิวเตอร์ทำงานได้เร็วและมีประสิทธิภาพ แต่นักเขียนโปรแกรมต้องมีความเข้าใจในโครงสร้างของระบบคอมพิวเตอร์อย่างลึกซึ้ง

สำหรับ**ภาษาระดับสูง**จะมีความใกล้เคียงกับภาษาอังกฤษ นักเขียนโปรแกรมเขียนได้ง่าย แต่โปรแกรมที่เขียนขึ้นต้องผ่านการแปลโดยโปรแกรมแปลภาษาให้เป็นภาษาเครื่องก่อน

ภาษาซีถูกสร้างขึ้นในช่วงปี พ.ศ. 2510 โดยนักคอมพิวเตอร์ ชื่อ นายเดนนิส ริชชี่ (Dennis Ritchie) จากห้องทดลองของเบลล์ (Bell Laboratories) ประเทศสหรัฐอเมริกา วัตถุประสงค์หลักในการพัฒนาคือ เพื่อใช้ในการสร้างระบบปฏิบัติการยูนิกส์ (UNIX) ที่เป็นต้นแบบของระบบปฏิบัติการตระกูลลินุกซ์ทั้งหลายในปัจจุบัน นอกจากนี้ภาษาซียังถูกนำไปใช้ในการสร้างโปรแกรมที่ทำงานเกี่ยวข้องโดยตรงกับระบบคอมพิวเตอร์ เช่น โปรแกรมประมวลคำ โปรแกรมตารางทำงาน โปรแกรมจัดการพื้นที่ของดิสก์ และโปรแกรมป้องกันและตรวจสอบไวรัส เนื่องจากภาษาซีมีความยืดหยุ่นสูง สามารถเขียนให้ใกล้เคียงกับภาษาระดับต่ำ ทำให้มีประสิทธิภาพในการทำงานสูง ผู้พัฒนาสามารถที่จะพัฒนาโปรแกรมได้โดยเน้นไปที่การแก้ปัญหาที่ต้องการได้อย่างอิสระโดยไม่ต้องคำนึงถึงฮาร์ดแวร์ใด ๆ

การเขียนโปรแกรมคอมพิวเตอร์นั้น จะต้องมีการประมวลผลกับข้อมูล โดยข้อมูลจะถูกเก็บในหน่วยความจำของคอมพิวเตอร์ในรูปแบบของตัวแปร การประกาศตัวแปรต่าง ๆ จะใช้หน่วยความจำไม่เท่ากัน และมีช่วงของการเก็บข้อมูลไม่เท่ากัน ผู้เขียนโปรแกรมจะต้องทราบว่าข้อมูลที่ต้องการประมวลผลนั้นเป็นข้อมูลประเภทใด และในการประมวลผลจะต้องมีการกระทำกับตัวแปรต่างๆ ตัวที่นำมากระทำเรียกว่า ตัวดำเนินการ ก่อนการเขียนโปรแกรมจึงต้องทำความเข้าใจกับองค์ประกอบต่าง ๆ ดังนี้

1. ตัวแปร

ตัวแปร (Variable) คือ การจองพื้นที่ในหน่วยความจำของคอมพิวเตอร์ สำหรับเก็บข้อมูลที่ต้องใช้ในการทำงานของโปรแกรม โดยมีการตั้งชื่อเรียกหน่วยความจำในตำแหน่งนั้นด้วย เพื่อความสะดวกในการเรียกใช้ข้อมูล ถ้าจะใช้ข้อมูลใดก็ให้เรียกผ่านชื่อของตัวแปรที่เก็บเอาไว้ หรืออาจหมายถึง การกำหนดตัวแปรที่เป็นการใช้ชื่อตัวแปรแทนตำแหน่งบนหน่วยความจำ สำหรับเก็บข้อมูลระหว่างการประมวลผล ซึ่งอาจเป็นข้อมูลนำเข้า ข้อมูลที่เกิดจากการดำเนินการ หรือข้อมูลผลลัพธ์

2. การตั้งชื่อตัวแปร

ผู้เขียนโปรแกรมจะต้องตั้งชื่อให้กับตัวแปร ค่าคงที่ โปรแกรมย่อย พารามิเตอร์ และส่วนต่างๆ ของโปรแกรม การตั้งชื่อให้กับตัวแปรจะเป็นไปตามหลักการตั้งชื่อของภาษาซี และชื่อที่เหมาะสมควรจะเป็นชื่อที่สื่อความหมาย กฎเกณฑ์ที่ใช้ในการตั้งชื่อของภาษาซี มีดังนี้

1. ชื่อจะต้องไม่ซ้ำกับคำสงวน (reserved word) และคำมาตรฐานที่คอมไพเลอร์รู้จัก
2. อักขระแรกของชื่อจะต้องเป็นตัวอักษร (A-Z,a-z) หรือเครื่องหมาย _ (underscore) เท่านั้น
3. ตัวต่อไปต้องเป็นตัวอักษร (A-Z,a-z) หรือตัวเลข หรือเครื่องหมาย _ (underscore)
4. ตัวพิมพ์ใหญ่และตัวพิมพ์เล็กถือว่าเป็นตัวอักษรคนละตัวกัน เช่น Salary และ SALARY ที่เป็นชื่อที่แตกต่างกัน เป็นต้น และต้องไม่มีช่องว่าง
5. ชื่อตามมาตรฐาน ANSI C จะมีความยาวไม่จำกัด แต่คอมไพเลอร์ตามมาตรฐาน ANSI C จะต้องสามารถจำแนกชื่อที่แตกต่างกันได้อย่างมาก 31 อักขระแรก

3. คำสงวน

คำสงวน หมายถึง คำที่สงวนไว้สำหรับเรียกใช้ตามวัตถุประสงค์ที่กำหนดไว้เฉพาะ เช่น คำที่ใช้ในคำสั่งควบคุม และชนิดของข้อมูล เป็นต้น

คำสงวนของภาษาซี (ANSI Standard C) ได้แก่

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void

default	goto	sizeof	volatile
do	if	static	while

4. การประกาศตัวแปรและค่าคงที่

4.1 การประกาศตัวแปร

การสร้างตัวแปรขึ้นมาใช้งาน เรียกว่า การประกาศตัวแปร สามารถทำได้ดังนี้

รูปแบบ **type variable_list; หรือ ประเภทข้อมูล <ชื่อตัวแปร...>;**

โดย type หมายถึง ชนิดข้อมูลของตัวแปร

variable หมายถึง ชื่อของตัวแปร

ในการประกาศตัวแปร สามารถประกาศครั้งละหลายตัวได้ ถ้าหากเป็นตัวแปรประเภทเดียวกันจะใช้เครื่องหมาย , คั่น ตัวอย่างเช่น ถ้าหากจะประกาศตัวแปรชื่อ Data1 และ Data2 สำหรับเก็บจำนวนเต็มสามารถทำได้ดังนี้

เช่น int Data1,Data2;

แต่ถ้าประกาศตัวแปรให้ชื่อ Data1 เก็บเลขจำนวนเต็ม และ Data2 เก็บเลขจำนวนจริง สามารถเขียนได้ดังนี้

เช่น int Data1;

float Data2;

4.2 การประกาศค่าคงที่

ค่าคงที่ (constant) เป็นค่าในหน่วยความจำที่มีค่าคงที่ตลอดโปรแกรม ในการประกาศค่าคงที่จะเป็นการกำหนดชื่อให้ค่าคงที่ ถ้าในโปรแกรมส่วนใดเรียกชื่อที่ประกาศไว้ก็จะได้ข้อมูลตามที่กำหนด สามารถทำได้ 2 ลักษณะคือ

รูปแบบที่ 1 const แบบข้อมูล ชื่อค่าคงที่ = ค่าข้อมูล;

เช่น const int count = 100;

เป็นการประกาศค่าคงที่ให้ชื่อว่า count ซึ่งเป็นข้อมูลแบบจำนวนเต็ม และเก็บค่า 100 เอาไว้

รูปแบบที่ 2 #define ชื่อค่าคงที่ ค่าที่ต้องการเก็บ

การประกาศค่าด้วย #define ไม่ต้องระบุประเภทของข้อมูล และไม่ต้องมีเครื่องหมาย ;

เช่น #define PI 3.14

เป็นการกำหนดค่าคงที่ให้ PI มีค่าเท่ากับ 3.14 เมื่อมีการประมวลผลให้ PI แทนด้วยเลขทศนิยมที่มีค่าเท่ากับ 3.14 การประกาศค่าคงที่ด้วยวิธีนี้ยังสามารถใส่เป็นนิพจน์ได้อีกด้วย

เช่น #define X (5+3) / 2

เป็นการประกาศค่าคงที่ให้ X มีค่าเท่ากับ 4

5. ตัวดำเนินการ

5.1 ตัวดำเนินการเลขคณิต

ใช้สำหรับกระทำการคำนวณทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร โดยจะนำข้อมูลตัวหนึ่งไปกระทำกับอีกตัวหนึ่ง โดยใช้ผลลัพธ์เป็นตัวเลขทางคณิตศาสตร์ ตัวดำเนินการทางคณิตศาสตร์ แบ่งออกได้ดังต่อไปนี้

ตัวดำเนินการ	กระบวนการ	ข้อมูลที่ถูกกระทำ	ข้อมูลผลลัพธ์
+	บวก(Addition)	จำนวนเต็ม,จำนวนจริง	จำนวนเต็ม,จำนวนจริง
-	ลบ(Subtraction)	จำนวนเต็ม,จำนวนจริง	จำนวนเต็ม,จำนวนจริง
*	คูณ(Multiplication)	จำนวนเต็ม,จำนวนจริง	จำนวนเต็ม,จำนวนจริง
/	หาร(real number Division)	จำนวนเต็ม,จำนวนจริง	จำนวนเต็ม,จำนวนจริง
%	การหารแบบเอาเศษ(Modulus)	จำนวนเต็ม	จำนวนเต็ม
++	การเพิ่มค่าขึ้นหนึ่ง(Increment)	จำนวนเต็ม	จำนวนเต็ม
--	การลดค่าลงหนึ่ง(Decrement)	จำนวนเต็ม	จำนวนเต็ม

ตัวอย่างที่ 1 การใช้ตัวดำเนินการแบบต่างๆ

```
3 + 4       =    7
7.0 - 3.0   =    4.0
6 * 1.5     =    9.0
9 / 2       =    4.5
9 % 2       =    1
```

ถ้ามีตัวดำเนินการหลายตัว ผลลัพธ์จะเกิดจากการกระทำของตัวดำเนินการแต่ละตัว ถ้าหากมีการใช้วงเล็บการกระทำใดๆ จะกระทำในวงเล็บก่อน สำหรับลำดับก่อนหลังของการทำตัวดำเนินการแสดงได้ดังตารางต่อไปนี้ โดยเรียงจากลำดับสูงสุดไปลำดับต่ำสุด

ตัวดำเนินการ	การทำงาน	ลำดับการทำงาน
( )	การทำในวงเล็บ	การทำงานในวงเล็บมีลำดับการทำงานสูงสุด
* / %	การคูณ การหาร DIV การหาร MOD	ถ้าหากมีตัวดำเนินการหลายตัวในประโยคเดียวกันจะทำจากซ้ายไปขวา
+ หรือ -	การบวก การลบ	ถ้าหากมีตัวดำเนินการหลายตัวในประโยคเดียวกันจะทำจากซ้ายไปขวา

คำถามที่ 1 ในการทำคำสั่งต่อไปนี้ ผลลัพธ์ที่ได้จะเป็นอย่างไร

ก. 1 + 2 * 3 + 4

ข. 6 + 4 / 2 + 3

ค. 2 / 3 * 4

คำถามที่ 2 พิจารณาโปรแกรมดังนี้ 2 + 3 * 4

จะมีค่าเท่ากับ 20 เพราะ (2 + 3) * 4

หรือเท่ากับ 14 เพราะ 2 + (3 * 4)

การใช้ตัวดำเนินการเพิ่มค่าและลดค่า (Increment and Decrement) ค่าของข้อมูลจะเปลี่ยนแปลงครั้งละหนึ่งค่า และใช้กับตัวแปรประเภทจำนวนเต็ม ตัวอย่างเช่น ถ้าหากต้องการเพิ่มค่าของตัวแปร x ขึ้นหนึ่งค่า อาจเขียนได้เป็น

```
x = x + 1;
```

ถ้าหากใช้ตัวดำเนินการเพิ่มค่าจะทำให้เขียนสั้นลง โดยเขียนได้ดังนี้

```
++x    หรือ    x++
```

และถ้าหากต้องการลดค่าในตัวแปร x ลงหนึ่งค่า อาจเขียนได้ดังนี้

```
- - x   หรือ    x - -
ซึ่งจะมีค่าเท่ากับ  x = x - 1;
```

ในการใช้ตัวดำเนินการเพิ่มค่าและลดค่านั้น การวางตัวดำเนินการไว้หน้าและหลังตัวแปรบางครั้งจะได้ผลไม่เท่ากัน เช่น ในกรณีที่ใช้ตัวดำเนินการแล้วส่งค่าให้กับอีกตัวแปรหนึ่ง ดังต่อไปนี้

```
x      =      10;
y      =      ++x;
```

จากการเขียนข้างบน เริ่มแรกให้ x มีค่าเท่ากับ 10 ต่อมาเพิ่มค่า x ขึ้นหนึ่งแล้วส่งให้ตัวแปร y หลังจากการทำประโยคข้างบนจะทำให้ตัวแปร y มีค่าเท่ากับ 11 แต่ถ้าเขียนเป็น

```
x      =      10;
y      =      x++;
```

หลังจากทำคำสั่งจะทำให้ตัวแปร y มีค่าเป็น 10 แต่ค่าใน x จะเป็น 11 ซึ่งสรุปได้ว่า **การวางตัวดำเนินการไว้ด้านหลังจะทำให้โปรแกรมส่งค่าให้กับตัวแปร y ก่อน จากนั้นเพิ่มค่าตัวมันขึ้นหนึ่ง แต่ถ้าวางตัวดำเนินการไว้ด้านหน้า โปรแกรมจะเพิ่มค่าให้กับตัวแปรก่อนจากนั้นจึงส่งให้กับตัวแปร y**

ตัวอย่างที่ 2 โปรแกรมพิมพ์ผลลัพธ์จากการใช้ตัวดำเนินการคำนวณมากกว่า 1 ตัว

```
/* 1 */   // Program : MultOper.cpp
/* 2 */   // พิมพ์ผลลัพธ์จากการใช้ตัวดำเนินการคำนวณมากกว่า 1 ตัว
/* 3 */
/* 4 */   #include<stdio.h>
/* 5 */
/* 6 */   main(){
/* 7 */   int a = 1 , b = 2 , c = 3 , d , e;
/* 8 */
/* 9 */     d = a * -b + c ;
/* 10 */     e  = a * ( -b + c) ;
/* 11 */     printf(" d is %d , e is %d .\n" , d , e);
/* 12 */
/* 13 */     d = a + b * c ;
/* 14 */     e = (a + b ) * c ;
/* 15 */     printf(" d is %d, e is %d.\n", d,e) ;
/* 16 */
/* 17 */     d = b % c + a ;
/* 18 */     e = b % ( c + a ) ;
/* 19 */     printf ( " d is %d, e is %d.\n", d,e) ;
/* 20 */
/* 21 */     d = c - b / a * a ;
/* 22 */     e = ( c - b ) / ( a * a ) ;
/* 23 */     printf(" d is %d, e is %d.\n", d,e);
/* 24 */
/* 25 */     printf (" 8 + 2 * 6 / 3 - 2 is %d.\n", 8 + 2 * 6 / 3 - 2);
/* 26 */     printf(" 5 %% 5 + 5 * 5 - 5 / 5 is %d.\n",5 % 5 + 5 * 5 - 5 / 5);
/* 27 */   }
```

ผลลัพธ์ คือ

```
d is 1 , e is 1.
d is 7 , e is 9.
d is 3 , e is 2.
d is 1 , e is 1.
8 + 2 * 6 / 3 - 2 is 10.
5 % 5 + 5 - 5 / 5 is 24.
```

5.2 ตัวดำเนินการเปรียบเทียบ (Relation Operators)

ตัวดำเนินการเปรียบเทียบ (Relation Operators) จะนำข้อมูลสองค่ามาเปรียบเทียบกัน โดยข้อมูลทั้งสองค่าจะต้องเป็นข้อมูลประเภทเดียวกัน ผลลัพธ์ที่ได้จะเป็นค่าทางลอจิก คือ จริงหรือเท็จ

ตัวดำเนินการ	กระบวนการ
==	เท่ากับ
!=	ไม่เท่ากับ
<=	น้อยกว่าหรือเท่ากับ
>=	มากกว่าหรือเท่ากับ
>	มากกว่า
<	น้อยกว่า

ตัวอย่างที่ 3 การใช้ตัวดำเนินการเปรียบเทียบ

```
6  >   2     ผลลัพธ์เป็นจริง เนื่องจาก 6 มากกว่า 2 จริง
8  >=  3     ผลลัพธ์เป็นจริง เนื่องจาก 8 มีค่ามากกว่าหรือเท่ากับ 3 จริง
5  >=  18    ผลลัพธ์เป็นเท็จ เนื่องจาก 5 ไม่ได้มากกว่าหรือเท่ากับ 18
7  ==  4     ผลลัพธ์เป็นเท็จ เนื่องจาก 7 ไม่เท่ากับ 4
```

ตัวอย่างที่ 4 โปรแกรมแสดงการเพิ่มและลดค่าตัวแปรในแบบย่อ

```
/* 1 */   // Program : IncDec.cpp
/* 2 */   // เพิ่มและลดค่าตัวแปรในแบบย่อ
/* 3 */
/*  4 */  #include<stdio.h>
/* 5 */
/* 6 */    main( ) {
/* 7 */       int  x  =  8;
/* 8 */
/* 9 */       printf( " x  is  %d . \n" , x );
/* 10 */      printf( " x++  is  %d . \n" , x++ );
/* 11 */      printf( " x  is  %d . \n\n" , x );
/* 12 */
/* 13 */      x  =  4 ;
/* 14 */      printf ( " x  is  %d . \n" , x ) ;
/* 15 */      printf ( " --x  is  %d . \n" , --x ) ;
/* 16 */      printf ( " x  is  %d . \n" , x ) ;
/* 17 */    }
```

ผลลัพธ์ คือ

```
x is 8.
x ++ is 8.
x is 9.


x is 4.
--x is 3.
x  is 3.
```

5.3 ตัวดำเนินการทางตรรกะ (Logical Operator)

ตัวดำเนินการทางตรรกะ (logical operator) ประกอบด้วย การทำ AND , OR และ NOT เมื่อกระทำกับค่าใด ผลลัพธ์ที่ออกมาจะเป็นจริงหรือเท็จ ตัวดำเนินการทางตรรกะแสดงได้ ดังตารางต่อไปนี้

ตัวดำเนินการ	การกระทำ
&&	AND ค่าสองค่า ถ้าค่าทั้งสองเป็นจริง ผลลัพธ์จะเป็นจริง
||	OR ค่าสองค่า ถ้าค่าทั้งสองเป็นเท็จ ผลลัพธ์จะเป็นเท็จ
!	เปลี่ยนค่า จากจริงเป็นเท็จ จากเท็จเป็นจริง

ตัวอย่างที่ 5 การใช้ตัวดำเนินการทางตรรกะ

การกระทำต่อไปนี้จะให้ผลลัพธ์เป็นจริง `(5 == 4 + 1) && (18 <= 6 * 4)`

ถ้าหากในประโยคภาษาซี มีการใช้ตัวดำเนินการเปรียบเทียบและตัวดำเนินการทางตรรกะหลายตัว โปรแกรมภาษาซีจะจัดลำดับความสำคัญการทำงานก่อนหลัง ดังต่อไปนี้

ตัวดำเนินการ	ลำดับการทำงาน
!	1
> >= < <=	2
== !=	3
&&	4
||	5

ตัวอย่างที่ 6 โปรแกรมพิมพ์ค่าของนิพจน์รูปแบบต่าง ๆ

```
/*  1 */   //  program: Expr.cpp
/*  2 */   //  พิมพ์ค่าของนิพจน์รูปแบบต่าง ๆ
/*  3 */
/*  4 */   #include<stdio.h>
/*  5 */
/*  6 */   main() {
/*  7 */      int   x = 1, y, z ;
/*  8 */
/*  9 */     printf( " value of 2*x=5 is % d.\n",  2 * x + 5 );
/*  10 */    printf( " value of assignment to x is % d. \n",x = 5);
/*  11 */    printf( " value of assignment to y is ");
/*  12 */    printf( " % d . \n",  y = 2 * x++ + 1 );
/*  13 */    printf( " x is % d and y is % d . \n",  x , y );
/*  14 */
/*  15 */    z = y = 4 * x + y ;
/*  16 */    printf ( " y is % d and z is % d. \n" , y , z );
/*  17 */    }
```

ผลลัพธ์ คือ

```
Value of 2*x+5 is 7.
Value of assignment to x is 5.
Value of assignment to y is 11.
x is 6 and y is 11.
y is 35 and z is 35.
```

ที่มา :

ธีรวัฒน์ ประกอบผล. (2552). **คู่มือการเขียนโปรแกรม ภาษา C.** กรุงเทพฯ : ซัคเซส มีเดีย.

สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี กระทรวงศึกษาธิการ. (2553). **หนังสือเรียนรายวิชาเพิ่มเติม เทคโนโลยีสารสนเทศและการสื่อสาร ภาษาซี ชั้นมัธยมศึกษาปีที่ 4-6.** กรุงเทพฯ: โรงพิมพ์ สกสค.