

วิชา การพัฒนาโปรแกรม (ภาษาซี)

รหัสวิชา ว31293

หน่วยการเรียนรู้ที่ 3 นิพจน์และตัวดำเนินการ

สาระการเรียนรู้

1. ตัวดำเนินการ (Operator)
2. ตัวถูกดำเนินการ (Operand)
3. นิพจน์ (Expression)
4. เครื่องหมายและตัวดำเนินการทางคณิตศาสตร์
5. การเปลี่ยนชนิดข้อมูล

ผลการเรียนรู้

มีความรู้ความเข้าใจเกี่ยวกับตัวดำเนินการและตัวถูกดำเนินการ นิพจน์ เครื่องหมายทางคณิตศาสตร์ และการเปลี่ยนชนิดข้อมูลในภาษาซีเพื่อการเขียนโปรแกรมได้

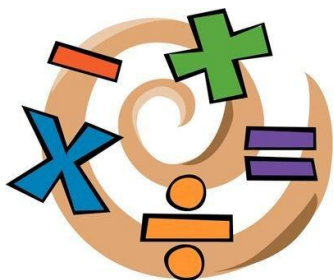
จุดประสงค์การเรียนรู้

1. บอกประวัติความเป็นมา และลักษณะของภาษา C ได้
2. สามารถค้นหาและดาวน์โหลดโปรแกรมภาษา C ได้
3. สามารถใช้งานโปรแกรมภาษา C ได้



ใบความรู้ที่ 3 เรื่อง การประมวลผลด้วยซี

จัดทำโดย นางพรพนรัตน์ ชมภูหุข



เมื่อรู้จักวิธีการเตรียมข้อมูลพร้อมทั้งการสร้างตัวแปรสำหรับเก็บข้อมูลขึ้นมาแล้ว ต่อไปจะเป็นการนำข้อมูลเหล่านั้นมาดำเนินการ (Operator) โดยการดำเนินการอาจจะหมายถึง การนำข้อมูลมาคำนวณทางคณิตศาสตร์ ตรรกศาสตร์ หรือเปรียบเทียบ เพื่อให้ได้ผลลัพธ์ออกมา ดังที่จะได้เรียนรู้กันในบทนี้



การประมวลผลด้วยภาษาซี

การเขียน โปรแกรมคอมพิวเตอร์นั้นต้องมีการประมวลผลข้อมูล โยข้อมูลจะถูกเก็บในหน่วยความจำคอมพิวเตอร์ในรูปแบบของตัวแปร การประกาศตัวแปรต่าง ๆ จะใช้หน่วยความจำไม่เท่ากัน และมีช่วงในการเก็บข้อมูลไม่เท่ากัน ผู้เขียนโปรแกรมจะต้องทราบว่าข้อมูลที่ต้องการประมวลผลนั้นเป็นข้อมูลประเภทใด และในการประมวลผลจะต้องมีการกระทำกับตัวแปรต่าง ๆ ดังนั้น ผู้เขียนโปรแกรมต้องทำความเข้าใจว่าการประมวลผลนั้นต้องรู้จักกับ ตัวดำเนินการ ตัวถูกดำเนินการ และนิพจน์ จึงจะสามารถเขียนโปรแกรมให้ทำงานตามที่ต้องการได้

1. ตัวดำเนินการ (Operator)

ตัวดำเนินการ คือ เครื่องหมายหรือสัญลักษณ์พิเศษที่ใช้ในการดำเนินการทางคณิตศาสตร์ การดำเนินการทางตรรกศาสตร์ หรืออื่น ๆ

2. ตัวถูกดำเนินการ (Operand)

ข้อมูลที่ทำงานกับตัวดำเนินการจะเรียกว่า ตัวถูกดำเนินการ ตัวดำเนินการบางตัวจะใช้ตัวถูกดำเนินการเพียงตัวเดียว ขณะที่ตัวดำเนินการบางตัวจะต้องใช้ตัวถูกดำเนินการถึง 2 ตัว และตัวดำเนินการจะใช้ประกอบกับตัวถูกดำเนินการในการสร้างนิพจน์

3. นิพจน์ (Expression)

นิพจน์ หมายถึง การนำค่าคงที่ หรือตัวแปร มาสัมพันธ์กันโดยใช้ตัวดำเนินการทางคณิตศาสตร์เป็นตัวเชื่อม ตัวอย่างเช่น

ตัวอย่าง



ดังนั้น นิพจน์ในภาษา C เป็นการนำข้อมูลและตัวแปรในภาษา C มาดำเนินการด้วยเครื่องหมายทางคณิตศาสตร์ ตรรกศาสตร์ หรือเครื่องหมายมาเปรียบเทียบกับภาษา C เป็นตัวสั่งงาน



เครื่องหมายและตัวดำเนินการทางคณิตศาสตร์

ในภาษาซีมีตัวดำเนินการ (Operator) อยู่หลายประเภท ซึ่งตัวดำเนินการเหล่านี้สามารถนำมาสร้างเป็นนิพจน์ได้ อันได้แก่

1. ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)
2. ตัวดำเนินการยูนารี (Unary Operator)
3. ตัวดำเนินการเชิงสัมพันธ์ (Relational Operator)
4. ตัวดำเนินการเชิงตรรกะ (Logical Operator)

1. ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)

การดำเนินการพื้นฐานที่สุดในชีวิตประจำวันและในการเขียนโปรแกรมก็คือ การคำนวณทางคณิตศาสตร์ ซึ่งถือได้ว่าเป็นการดำเนินการที่ต้องกระทำอยู่บ่อยครั้ง โดยเครื่องหมายที่ใช้ในการคำนวณทางคณิตศาสตร์ในภาษา C มีดังนี้

ตัวดำเนินการ	ความหมาย
+	บวก (addition)
-	ลบ (subtraction)
*	คูณ (multiplication)
/	หาร (division)
%	หารเอาเศษ (Modulo)

ตัวอย่าง ถ้ากำหนดให้ตัวแปร x และ y เก็บค่าจำนวนเต็มเท่ากับ 15 และ 9 ตามลำดับ ผลลัพธ์ที่ได้คือ

$z=x+y$	ผลลัพธ์ที่ได้คือ ตัวแปร z มีค่าเท่ากับ 24
$z=x-y$	ผลลัพธ์ที่ได้คือ ตัวแปร z มีค่าเท่ากับ 6
$z=x*y$	ผลลัพธ์ที่ได้คือ ตัวแปร z มีค่าเท่ากับ 135
$z=x/y$	ผลลัพธ์ที่ได้คือ ตัวแปร z มีค่าเท่ากับ 1
$z=x\%y$	ผลลัพธ์ที่ได้คือ ตัวแปร z มีค่าเท่ากับ 6

2. ตัวดำเนินการยูนิารี (Unary Operator)

ตัวดำเนินการยูนิารีจะใช้กับตัวถูกดำเนินการเพียงตัวเดียว ปกติจะอยู่หน้าตัวถูกดำเนินการ แต่บางครั้งก็อาจอยู่หลังตัวถูกดำเนินการ ตัวดำเนินการยูนิารีที่รู้จักกันดี คือ ตัวดำเนินการยูนิารีลบ นั่นก็คือเครื่องหมายลบที่นำมาใช้นำหน้าค่าตัวเลข หรือนำหน้าค่าตัวแปร ซึ่งจะส่งผลให้ค่าถูกเปลี่ยนเป็นค่าติดลบ โดยทันที เช่น -10 , $-x$

ตัวอย่าง ถ้ากำหนดให้ตัวแปร x และ y เก็บค่าจำนวนเต็มเท่ากับ 10 และ 2 ตามลำดับ ผลลัพธ์ที่ได้คือ

$z=x+y$	ผลลัพธ์ที่ได้คือ ตัวแปร z มีค่าเท่ากับ 12
$z=-x+y$	ผลลัพธ์ที่ได้คือ ตัวแปร z มีค่าเท่ากับ -8
$z=-x*y$	ผลลัพธ์ที่ได้คือ ตัวแปร z มีค่าเท่ากับ -20
$z=x- y$	ผลลัพธ์ที่ได้คือ ตัวแปร z มีค่าเท่ากับ 12



กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมทดสอบนิพจน์ยูนิารีต่อไปนี้ สังเกตผลลัพธ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 main ()
4 {
5     int a=10, b=2;
6     printf("a=10, b=2 \n \n");
7     printf("-a+b = %d \n ", -a+b);
8     printf("-a*b = %d \n ", -a*b);
9     printf("a--b = %d \n ", a- b);
10    getch();
11 }
```

รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex04_1.exe

นอกจากนี้ยังมีตัวดำเนินการยูนิารีที่ใ้้มาก ก็คือ ตัวดำเนินการเพิ่มค่าหรือ ลดค่า ซึ่งการใช้เครื่องหมายทางคณิตศาสตร์ประเภทของการเพิ่มค่าลดค่าลงทีละหนึ่ง สามารถใช้ได้ดังแสดงในตาราง

ตัวดำเนินการ	ความหมาย
++	เพิ่มค่าขึ้นอีกหนึ่ง (increment)
--	ลดค่าลงอีกหนึ่ง (decrement)

การเขียนยูนิารีแบบเพิ่มค่าหรือลดค่า แบบเพิ่มค่าหรือลดค่า สามารถกำหนดได้ 2 รูปแบบคือ

แบบที่ 1 เครื่องหมายอยู่ข้างหน้า

ตัวอย่าง $z = ++x$
หมายถึง เพิ่มค่าให้กับตัวแปร x อีก 1 ก่อนที่กำหนดค่าให้กับตัวแปร z

แบบที่ 2 เครื่องหมายอยู่ข้างหลัง

ตัวอย่าง $z = x++$
หมายถึง กำหนดค่าของตัวแปร x ให้กับตัวแปร z ก่อนที่จะเพิ่มค่าให้กับตัวแปร x อีก 1



กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมทดสอบนิพจน์ยูนิารีต่อไปนี้ สังเกตผลลัพธ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 int x=5 , y=5 , z=0;
4 main( )
5 {
6     z=++x;
7     printf("z = %d \n", z);
8     printf("x = %d\n", x);
9     z=y++;
10    printf("z = %d \n", z);
11    printf("y = %d\n", y);
12    getch();
13 }
```

```

C:\ D:\stc6107\ex04\ex04_2.exe
z=6
x=6
z=5
y=6
```

รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของ
โปรแกรมex04_2.exe

นอกจากนี้แล้วยังมีตัวดำเนินการอีกประเภทหนึ่งคือ ประเภทลดรูป ซึ่งเรียกว่า Compound Assignment ซึ่งได้แก่ตัวดำเนินการต่อไปนี้

ตัวดำเนินการ	ตัวอย่าง	ความหมายเหมือนกัน
$+=$	$a+=b;$	$a= a+b;$
$-=$	$a-=b;$	$a= a-b;$
$*=$	$a*=b;$	$a= a*b;$
$/=$	$a/=b;$	$a= a/b;$
$\%=$	$a\%=b;$	$a= a\% b;$

ตัวอย่าง บวกแบบลดรูป

$a+=5;$ หรืออีกความหมาย คือ $a=a+5;$

หมายถึง บวกค่าในตัวแปร z กับ 5 แล้วนำผลลัพธ์กลับไปเก็บไว้ที่ตัวแปร z อีกครั้ง

ตัวอย่าง ลบแบบลดรูป

$a-=2;$ หรืออีกความหมาย คือ $a=a-2;$

หมายถึง ลบค่าในตัวแปร a ด้วย 2 แล้วนำผลลัพธ์กลับไปเก็บไว้ที่ตัวแปร a อีกครั้ง

ตัวอย่าง คูณแบบลดรูป

$a*=b;$ หรืออีกความหมาย คือ $a=a*b;$

หมายถึง คูณค่าในตัวแปร z กับค่าในตัวแปร x แล้วนำผลลัพธ์กลับไปเก็บไว้ที่ตัวแปร z อีกครั้ง

ตัวอย่างหารแบบลดรูป

$a/=4;$ หรืออีกความหมาย คือ $a=a/4;$

หมายถึง หารค่าในตัวแปร a ด้วย 4 แล้วนำผลลัพธ์กลับไปเก็บไว้ที่ตัวแปร a อีกครั้ง

ตัวอย่าง หารเอาเศษแบบลดรูป

$a \% = b$; หรืออีกความหมาย คือ $a = a \% b$;

หมายถึง หาค่าในตัวแปร a กับค่าในตัวแปร b แล้วผลลัพธ์ที่ได้จากการหารกลับไปเก็บไว้ที่ตัวแปร a อีกครั้ง

3. ตัวดำเนินการเชิงสัมพันธ์ (Relational Operator)

ตัวดำเนินการเชิงสัมพันธ์ เป็นเครื่องหมายที่ใช้ในการเปรียบเทียบ ตัดสินใจ และผลของการเปรียบเทียบเป็นได้ 2 กรณี คือ เป็นจริง (True) หรือ เป็นเท็จ (False) ถ้าเป็นจริงจะให้ค่าเป็น 1 และถ้าเป็นเท็จ จะให้ค่าเป็น 0 ตัวดำเนินการเชิงสัมพันธ์ที่ใช้มีดังนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
<	น้อยกว่า	$a < b$
<=	น้อยกว่าเท่ากับ	$a <= b$
>	มากกว่า	$a > b$
>=	มากกว่าเท่ากับ	$a >= b$
==	เท่ากับ	$a == b$
!=	ไม่เท่ากับ	$a != b$

ส่วนใหญ่แล้วการดำเนินการเปรียบเทียบจะทำงานร่วมกับการดำเนินการอื่น ๆ เช่น เปรียบเทียบจากการคำนวณทางคณิตศาสตร์ หรือเปรียบเทียบเพื่อกำหนดเงื่อนไขร่วมกับการทำงานของโปรแกรมโครงสร้างแบบเลือกทำ (Selection) หรือแบบทำซ้ำ เช่น ในร่วมกับคำสั่ง if หรือคำสั่ง while เป็นต้น

ตัวอย่าง ถ้า a , b และ c เป็นตัวแปรจำนวนเต็มมีค่าเป็น 10, 20 และ 30 ตามลำดับ เมื่อกำหนดรูปแบบการทำงานดังในตาราง

ตัวดำเนินการ	ผล	ค่าที่ได้
$a < b$	จริง	1
$(b+c) > (a+50)$	เท็จ	0
$c != 30$	เท็จ	0



กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียนโปรแกรมทดสอบนิพจน์ยูนิารีต่อไปนี้ สังเกตผลลัพธ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 main()
4 {
5     printf("10 > 5 = %d\n", 10 > 5);
6     printf("10 < 5 = %d\n", 10 < 5);
7     getch();
8 }

```

รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของ
โปรแกรม ex04_3.exe

4. ตัวดำเนินการเชิงตรรกะ (Logical Operator)

ตัวดำเนินการทางตรรกะ เป็นเครื่องหมายที่ใช้ในการเปรียบเทียบตัดสินใจ โดยเอาเงื่อนไขตั้งแต่ 2 เงื่อนไขขึ้นไปมาเปรียบเทียบกัน ผลของการเปรียบเทียบเป็นได้ 2 กรณีเป็นจริง (True) หรือ เป็นเท็จ (False) ถ้าเป็นจริงจะให้ค่าเป็น 1 และถ้าเป็นเท็จ จะให้ค่าเป็น 0 ตัวดำเนินการทางตรรกะที่ใช้มีดังนี้

ตัวดำเนินการ	ความหมาย
&&	และ (and)
	หรือ (or)
!	ปฏิเสธ (not)

โดยผลลัพธ์สามารถแสดงเป็นตารางแสดงการเปรียบเทียบผลลัพธ์ของการเปรียบเทียบตัดสินใจได้ดังนี้

ตัวเปรียบเทียบ		ผลลัพธ์		
a	b	a && b	a b	!a
T	T	T	T	F
T	F	F	T	F
F	F	F	T	T
F	T	F	T	T



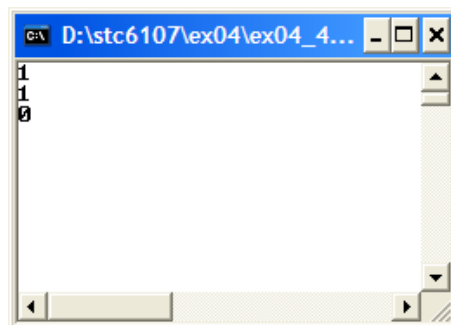
กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียน โปรแกรมทดสอบนิพจน์ยูนิอาร์ต่อไปนี้ สังเกตผลลัพธ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 main()
4 {
5     int score=85;
6     printf(“%d\n”,score>=80 && score<90);
7     printf(“%d\n”,score< 80 || score>50);
8     printf(“%d\n”,score==80);
    getch();
}

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของ
โปรแกรม ex04_4.exe



ตัวดำเนินการกับความสำคัญ

ตัวดำเนินการแต่ละตัวมีลำดับความสำคัญก่อนหลังที่ต่างกัน โดยการประมวลผลจะกระทำกับตัวดำเนินการที่มีลำดับความสำคัญสูงก่อน แต่ถ้ากรณีที่มีลำดับความสำคัญเท่ากัน ตามปกติจะกระทำกับตัวดำเนินการที่พบก่อนนั่นเอง และต่อไปนี้เป็นลำดับความสำคัญของตัวดำเนินการพื้นฐาน

ลำดับความสำคัญ	ตัวดำเนินการ	ความหมาย
1	()	เครื่องหมายวงเล็บ
2	++,--	ตัวดำเนินการเพิ่มค่า/ลดค่า
3	-,!	ยูนิอาร์ลอป และตรรกะ not
4	*,/,%	คูณ หหาร โมดูลัส
5	+, -	บวก ลบ

ตัวอย่าง การคำนวณนิพจน์ สามารถลำดับการคำนวณ ได้ดังนี้

$$\begin{aligned}
 \text{โจทย์} \quad \text{ans} &= 2 + 8 / 4 - 12 / 3 + 8 * 2 \\
 &= 2 + \underbrace{8 / 4} - \underbrace{12 / 3} + \underbrace{8 * 2} \\
 &= 2 + 2 - 4 + 16 \\
 \text{ans} &= 16
 \end{aligned}$$

ท้ายสุดผลลัพธ์ที่ได้ก็คือ 8 อย่างไรก็ตามการคำนวณสามารถเขียนให้ดูง่ายขึ้น และเพื่อป้องกันการคำนวณที่ผิดพลาดได้ ด้วยการใส่เครื่องหมาย () กำกับ โดยการคำนวณจะกระทำในแต่ละวงเล็บให้เสร็จก่อน

ตัวอย่าง

$$\text{income} = \text{salary} + (\text{ot} * \text{rate}) + \text{commission} - \text{tax}$$


การเปลี่ยนชนิดข้อมูล

ในภาษา C ยังมีตัวดำเนินการที่เรียกว่า การแคสต์ (Casting) เพื่อแปลงชนิดข้อมูลจากชนิดหนึ่ง มาเป็นอีกชนิดหนึ่งได้ วิธีทำ คือ ให้ระบุชนิดข้อมูลที่ต้องการภายในเครื่องหมายวงเล็บ หน้านิพจน์ที่ต้องการ

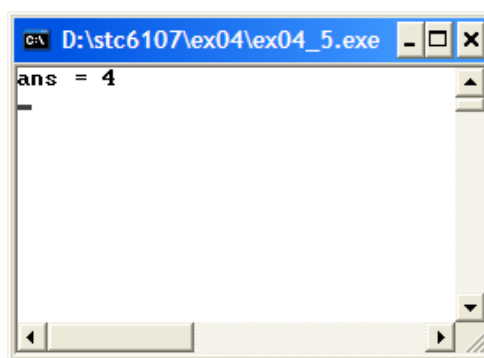


กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียน โปรแกรมทดสอบการเปลี่ยนชนิดข้อมูลต่อไปนี้ แล้วสังเกตผลลัพธ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 float x = 20;
4 int ans;
5 main()
6 {
7     ans = (int) x / 5;
8     printf("ans = %d\n",ans);
9     getch();
10 }
```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรม ex04_5.exe



การเขียนโปรแกรมคำนวณ

เราสามารถคำนวณหาผลลัพธ์ของนิพจน์คณิตศาสตร์ด้วยการเขียนโปรแกรมภาษาซี ตัวอย่างต่อไปนี้เป็น การแสดงลำดับการประมวลผลทางคณิตศาสตร์ของนิพจน์ต่าง ๆ



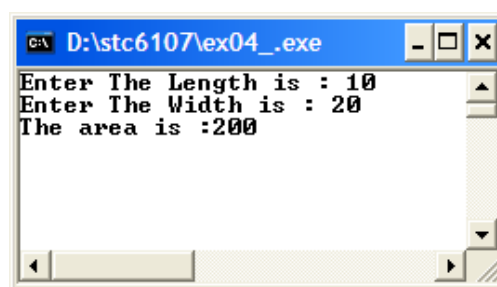
กิจกรรมเสริมทักษะ

การเขียนโปรแกรมคำนวณหาพื้นที่รูปสี่เหลี่ยม ให้นักเรียนทดลองเขียนโปรแกรมทดสอบการเปลี่ยนชนิดข้อมูลต่อไปนี้ แล้วสังเกตผลลัพธ์ที่ได้

```

1 #include <stdio.h>
2 #include <conio.h>
3 int x,y,sum;
4 main()
5 {
6     printf("Enter The Length is : ");
7     scanf ("%d", &x);
8     printf("Enter The Width is : ");
9     scanf ("%d", &y);
10    sum = x*y;
11    printf("The area is : %d", sum);
12    getche();
}

```



รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของ
โปรแกรม ex04_6.exe



กิจกรรมเสริมทักษะ

ให้นักเรียนทดลองเขียน โปรแกรมทดสอบการเปลี่ยนชนิดข้อมูลต่อไปนี้ แล้วสังเกตผลลัพธ์ที่ได้

```

1  #include <stdio.h>
2  #include <conio.h>
3  main( )
4  {
5      int a,b,c,d,e;
6      a=(3+4)*5;
7      b=3+4*5;
8      c=(2+7)*4%10;
9      d=2+7*4%10;
10     e=10+2*8/4*3-5;
11     printf("(3 + 4) * 5 =%d \n" , a);
12     printf("3 + 4 * 5 =%d \n" , b);
13     printf("(2 + 7) * 4 % 10 =%d \n" , c);
14     printf("(2 + 7) * 4 % 10 =%d \n" , d);
15     printf("10 + 2 * 8 / 4 * 3 - 5 =%d \n" , e);
16     getch();
17 }

```

```

D:\stc6107\ex04_7.exe
(3+4)*5 =35
3+4*5 =23
(2+7)*4 =6
(2+7)*4 =10
10+2*8/4*3-5 =17

```

รูปภาพ แสดงผลลัพธ์ที่ได้จากการทำงานของ
โปรแกรม ex04_7.exe