หน่วยเรียนที่ 3

ตัวดำเนินการและนิพจน์

การเขียนโปรแกรมโดยส่วนใหญ่จะเกี่ยวข้องกับตัวเลขและตัวดำเนินการทางคณิตศาสตร์ ซึ่งต้องมีองค์ประกอบที่สำคัญนั่นคือ ตัวดำเนินการ และตัวแปร มากระทำร่วมกันเป็นนิพจน์ในลักษณะต่างๆ ตัวดำเนินการที่สำคัญได้แก่ ตัวดำเนินการกำหนดค่า ตัวดำเนินการทางคณิตศาสตร์ต่างๆ เช่นการคูณ หาร บวก ลบ และตัวดำเนินการตรรกะ เป็นต้น

3.1 ตัวดำเนินการกำหนดค่า

ตัวดำเนินการกำหนดค่า เป็นตัวดำเนินการพื้นฐานที่ใช้ในการกำหนดค่าต่างๆ ให้กับตัวแปร โดยใช้เครื่องหมาย = มีรูปแบบของนิพจน์ในการกำหนดค่า คือ

ตัวแปร = นิพจน์

โดยที่ นิพจน์อาจจะเป็นค่าคงที่ ตัวแปร หรือนิพจน์ที่ประกอบขึ้นจากตัวดำเนินการต่างๆ โดยค่าที่อยู่ทางขวาของเครื่องหมาย = เป็นค่า หรือผลลัพธ์ให้แก่ตัวแปรทางซ้ายมือ เช่น

```
Num1 = 10;
Num2 = 20;
Num3 = (Num1 * Num2)/ 2 ;
```

นอกจากนี้ยังมีรูปแบบเพิ่มเติมหากต้องการกำหนดค่าให้กับตัวแปรหลายตัวด้วยค่าเดียวกัน เช่น

```
int x,y,z;
x = y = z = 35;
```

นอกจากนี้เมื่อมีการใช้ตัวดำเนินการกำหนดค่าจะมีการแปลงค่าชนิดข้อมูลโดยอัตโนมัติอีกด้วย (Implicit Casting) ซึ่งจะเกิดกับข้อมูลตัวเลข เช่น กำหนดค่าที่มีชนิดเป็นตัวเลข

จำนวนเต็ม int ให้กับตัวแปรชนิดข้อมูลจำนวนจริง float จะเกิดกระบวนการแปลงค่าจากจำนวนเต็ม int ไปเป็นข้อมูลชนิดจำนวนจริง float เช่น

```
float x;
x = 18;
```

โดยที่ 18 เป็นข้อมูลชนิดจำนวนเต็ม int แต่ตัวแปร x เป็นตัวแปรชนิดจำนวนจริง float เพราะฉะนั้นค่า 18 จะถูกแปลงเป็น 18.0f ก่อนที่จะนำไปเก็บยังตัวแปร x แต่หากมีการกำหนดค่าที่เป็นชนิดจำนวนจริง ให้กับตัวแปรชนิดจำนวนเต็ม จะเกิดการปัดเศษ ทิ้งก่อนนำไปเก็บยังตัวแปร เช่น

```
int i;
i = 4.5;
```

จะมีการแปลงค่า 4.5 เป็น 4 ก่อน หรือเกิดการปัดเศษทิ้ง ก่อนจะนำค่าไปกำหนดให้แก่ตัวแปร i

ตัวอย่างที่ 3.1 การกำหนดค่าจำนวนจริงให้แก่ตัวแปรจำนวนเต็ม

```
#include <stdio.h>
void main()
{
  int x;
  x = 15.3426;
  printf("x value is %d",x);
}
```

จะได้ผลลัพธ์ของโปรแกรมดังภาพที่ 3.1

```
x value is 15
```

ภาพที่ 3.1 ผลลัพธ์บนจอภาพของโปรแกรมตัวอย่างที่ 3.1

3.2 ตัวดำเนินการทางคณิตศาสตร์

การดำเนินการทางคณิตศาสตร์ของภาษีซีนั้นจะเกี่ยวข้องกับการคำนวณทางคณิตศาสตร์พื้นฐานโดยใช้ตัวดำเนินการ คือการบวก การลบ การคูณ การหาร โดยจะใช้เครื่องหมาย +, - , *, / ตามลำดับ กระทำระหว่างตัวเลขกับตัวเลข ตัวแปรกับตัวเลข และตัวแปรกับตัวแปร ซึ่งสรุปลักษณะตัวดำเนินการไว้ในตารางที่ 3.1 ทั้งนี้หากในหนึ่งสมการมีนิพจน์หลายนิพจน์จะให้ความสำคัญกับนิพจน์ที่อยู่ในวงเล็บก่อน

ตารางที่ 3.1 ตัวดำเนินการต่างๆทางคณิตศาสตร์

ตัวดำเนินการ	ชื่อตัวดำเนินการ	ตัวอย่าง	ผลลัพธ์
+	บวก	a+b	35
-	ลบ	b-a	15
*	คูณ	a*b	250
/	หาร(เอาเฉพาะจำนวนเต็ม)	b/a	2
%	Mod (หารเอาเฉพาะเศษ)	b%a	5
-	นิเสธ	-b	-25

หมายเหตุ กำหนดให้ตัวแปร a = 10, b = 25

ตัวดำเนินการทางคณิตศาสตร์จะมีลำดับความสำคัญไม่เท่ากัน เหมือนกับระบบคณิตศาสตร์ทั่วไป นั่นคือ เครื่องหมาย *,/, % จะมีลำดับความสำคัญมากกว่าเครื่องหมาย +, - หากพบตัวดำเนินการที่อยู่ในลำดับเดียวกันก็จะแปลความหมายให้ความสำคัญกับการคูณ การหารก่อน โดยลำดับความสำคัญของเครื่องหมายที่กระทำกัน เช่น

3 + 4/2 ; จะได้ความหมายเท่ากับ 3 + (4/2); = 5

3 * 2 + 4 % 2 ; จะได้ความหมายเท่ากับ (3*2) + (4%2); = 6 + 0 = 6

แต่ถ้าหากต้องการให้กระทำตัวดำเนินการลำดับต่ำก่อนให้ใช้เครื่องหมาย ( ) ครอบนิพจน์ที่ต้องการ เช่น

3 * (2 + 4) % 2 จะได้ความหมายเท่ากับ (3 * 6) % 2; = 18 %2 = 0

ดังนั้นหากต้องการเขียนคำสั่งที่มีสมการเป็น

$$x = \frac{a + b - c}{20}$$

ต้องเขียนเป็นคำสั่งเป็นนิพจน์ทางคณิตศาสตร์ดังนี้

```
x = ( a + b - c) /20 ;
```

ตัวอย่างที่ 3.2 โปรแกรมหาผลรวมของเลขจำนวนเต็ม 2 จำนวน ที่รับข้อมูลจากผู้ใช้

```
#include <stdio.h>
void main(void)
{
  int x,y,z;
  printf("Enter X value :");
  scanf("%d",&x);
  printf("\n Enter Y value :");
  scanf("%d ",&y);
  z = x + y ;
  printf("\n Summary of X and Y is %d",z);
}
```

ผลการทำงานของโปรแกรมเป็นดังภาพที่ 3.2

```
Enter X value : 15
Enter Y value : 20
Summary of X and Y is 35
```

ภาพที่ 3.2 ผลลัพธ์บนจอภาพของโปรแกรมตัวอย่างที่ 3.2

3.3 ตัวดำเนินการกำหนดค่าแบบผสม

ตัวดำเนินการกำหนดค่าแบบผสม เป็นตัวดำเนินการที่ผสมระหว่างตัวดำเนินการกำหนดค่า และตัวดำเนินการคณิตศาสตร์แบบย่อ เพื่อให้การเขียนโปรแกรมสั้นและกระชับมากยิ่งขึ้น โดยลักษณะของตัวดำเนินการกำหนดค่าแบบผสมดังในตารางที่ 3.2

ตารางที่ 3.2 ตัวดำเนินการกำหนดค่าแบบผสม

ตัวดำเนินการ	ตัวอย่างคำสั่ง	คำสั่งเต็ม
*=	x *= 5;	x = x*2;
/=	x /= 5;	x = x/5;
%=	x %= 5;	x = x%5;
+=	x += 5;	x = x+5;
-=	x -= 5;	x = x-5;

การแปลคำสั่งของตัวดำเนินการกำหนดค่าแบบผสม จะทำคำสั่งที่อยู่ทางขวามือก่อนเสมอ เช่น

x *= 5 + 3; จะเท่ากับคำสั่ง x = x *(5+3);

3.4 ตัวดำเนินการเพิ่มและลดค่า

ตัวดำเนินการเพิ่มและลดค่า เป็นตัวดำเนินการเพื่อใช้เพิ่มค่าตัวแปรขึ้น 1 ค่า หรือเป็นการลดค่าตัวแปรลง 1 ค่า โดยใช้เครื่องหมาย ++ แทนการเพิ่มค่าขึ้น 1 และ ใช้เครื่องหมาย - - แทนการลดค่าตัวแปรลง 1 ทั้งนี้ตำแหน่งของเครื่องหมายอาจจะวางไว้หน้าตัวแปร หรือหลังตัวแปรก็ได้ ซึ่งจะมีความหมายเหมือนกัน เช่น

x++; หรือ ++x; จะมีค่าเท่ากับ x = x+1; และ

x- -; หรือ - -x; จะมีค่าเท่ากับ x = x-1;

ความแตกต่างของตำแหน่งการวางเครื่องหมายด้านหน้าหรือหลังตัวแปร เมื่อเป็นนิพจน์ที่ซับซ้อนขึ้น เช่น

```
x = 5;
y = 10 + x++ * 4;
```

ผลลัพธ์ที่ได้จะมีผลเท่ากับคำสั่งดังนี้

```
y = 10 + (x*4);
x = x+1;
```

ซึ่งหมายความว่า y จะมีค่าเท่ากับ 10 + 5*4 = 30 และ x เท่ากับ 6 แต่หากนำตัวดำเนินการมาไว้ด้านหน้าตัวแปร ความหมายจะเปลี่ยนไป เช่น

```
x = 5;
y = 10 + ++x * 4;
```

ผลลัพธ์ที่ได้จะมีผลเท่ากับคำสั่งดังนี้

```
x = x+1;
y = 10 + (x*4);
```

ซึ่งหมายความว่าจะเพิ่มค่าให้ x เท่ากับ 6 ก่อน แล้ว y จะมีค่าเท่ากับ 10 + 6*4 = 34

ตัวอย่างที่ 3.3 การใช้ตัวดำเนินการเพิ่มค่า

```
#include <stdio.h>
void main()
{
  int x,y;
  x= 1;
  y = x++;
  printf(" y = %d\n ",y);
  y = ++x;
  printf("y = %d\n",y);
}
```

ผลการทำงานของโปรแกรมเป็นดังภาพที่ 3.3

```
y = 2
y = 3
```

ภาพที่ 3.3 ผลลัพธ์บนจอภาพของโปรแกรมตัวอย่างที่ 3.3

3.5 ตัวดำเนินการเชิงเปรียบเทียบ

ตัวดำเนินการเปรียบเทียบ (comparator) จะใช้สำหรับเปรียบเทียบจำนวนสองจำนวน ซึ่งจะให้ผลลัพธ์ออกมา 2 ค่า คือค่าเป็นจริง (True) หรือ เท็จ (False) เท่านั้น หากกำหนดให้ตัวแปร a = 10, b = 25 ผลการเปรียบเทียบแบบต่างๆ จะให้ผลลัพธ์ดังในตารางที่ 3.3

ตารางที่ 3.3 ตัวดำเนินการต่างๆเชิงเปรียบเทียบ

ตัวดำเนินการ	ชื่อตัวดำเนินการ	ตัวอย่าง	ผลลัพธ์
<	น้อยกว่า	a < b	true หรือมีค่าเท่ากับ 1
<=	น้อยกว่าหรือเท่ากับ	a <= b	true หรือมีค่าเท่ากับ 1
>	มากกว่า	a > b	false หรือมีค่าเป็น 0
>=	มากกว่าหรือเท่ากับ	a >= b	false หรือมีค่าเป็น 0
==	เท่ากัน	a == b	false หรือมีค่าเป็น 0
!=	ไม่เท่ากัน	a != b	true หรือมีค่าเท่ากับ 1

ตัวอย่างที่ 3.4 แสดงค่าผลการเปรียบเทียบ

```
#include <stdio.h>
void main()
{
  int x,y;
  printf("Please Enter X =");
  scanf("%d",&x);
  printf("\n Please Enter Y =");
  scanf("%d",&y);
  printf("\n X > Y is  %d",x>y);
}
```

ผลการทำงานของโปรแกรมเป็นดังภาพที่ 3.4

```
Please Enter X = 30
Please Enter Y = 25
X > Y is 1
```

ภาพที่ 3.4 ผลลัพธ์บนจอภาพของโปรแกรมตัวอย่างที่ 3.4

3.6 ตัวดำเนินการตรรกะ

ตัวดำเนินการตรรกะเป็นตัวดำเนินการที่ใช้คู่กับตัวดำเนินการเชิงเปรียบเทียบ มักจะใช้ร่วมกับคำสั่งควบคุมที่มีการกำหนดเงื่อนไข อย่างน้อย 2 เงื่อนไข และนำผลลัพธ์ของแต่ละเงื่อนไขมากระทำทางตรรกะ แล้วจะได้ผลลัพธ์เป็นจริง หรือเป็นเท็จเช่นกันเดียวกัน สมมติหากกำหนดให้ตัวแปร a = 10, b = 25 ผลจากตัวดำเนินการตรรกะจะให้ผลลัพธ์ดังในตารางที่ 3.4

ตารางที่ 3.4 ตัวดำเนินการตรรกะ

ตัวดำเนินการ	ชื่อตัวดำเนินการ	ตัวอย่าง	ผลลัพธ์
&&	และ(Logical AND)	(a>b) && (b<a)	false หรือมีค่าเป็น 0
||	หรือ (Logical OR)	(a>b) || (15>a)	true หรือมีค่าเท่ากับ 1
!	ไม่ใช่	!(30<a)	true หรือมีค่าเท่ากับ 1

3.7 ตัวดำเนินตรรกะระดับบิต

การดำเนินการตรรกะในระดับบิต เพื่อให้การทำความเข้าใจได้ง่ายขึ้นจะต้องแปลงข้อมูลให้เป็นเลขฐานสองเสียก่อน แล้วดำเนินการระหว่างบิตต่อบิตของแต่ละตำแหน่งของข้อมูล ซึ่งเป็นการกระทำทางดิจิตอล เช่นการ แอนด์ (AND) หรือ ออร์ (OR) เช่น

a = 10 แปลงเป็นเลขฐานสองขนาด 8 บิต คือ 0000 1010

b = 25 แปลงเป็นเลขฐานสองขนาด 8 บิต คือ 0001 1001

หาก a & b = 0000 1010 &

0001 1001

0000 1000 (มีค่าเท่ากับ 8_{10})

ส่วนผลการดำเนินตรรกะระดับบิตอื่นๆ ผลลัพธ์ดังตารางที่ 3.5

ตารางที่ 3.5 ตัวดำเนินการตรรกะระดับบิต

ตัวดำเนินการ	ชื่อตัวดำเนินการ	ตัวอย่าง	ผลลัพธ์
&	แอนด์	b&a	0000 1000 (8_{10})
|	ออร์	b|a	0001 1011 (27_{10})
^	เอ็กคลูซีฟ-ออร์	b^a	0001 0011 (19_{10})
<<	shift left	b<<2	0110 0100 (100_{10})
>>	shift right	b>>3	0000 0011 (3_{10})
~	Complement	~a	1111 0101 (245_{10})

3.8 ตัวดำเนินการเงื่อนไข

ตัวดำเนินการเงื่อนไขเป็นตัวดำเนินการที่ใช้ตรวจสอบเงื่อนไขของนิพจน์ โดยมีรูปแบบดังนี้

นิพจน์1 ? นิพจน์2 : นิพจน์3

การดำเนินการจะมีการตรวจสอบนิพจน์1 ว่าเป็นจริงหรือเท็จ หากเป็นจริงจะทำงานในนิพจน์2 แต่หากเป็นเท็จจะทำงานในนิพจน์3 เช่น

```
y  = (a >b)? a : b;
```

หมายความว่า จะมีการเปรียบเทียบค่าในตัวแปร a และ b ก่อน หากตัวแปร a มากกว่าตัวแปร b จริง จะทำให้ y เท่ากับ a แต่หากเป็นเท็จ y จะเท่ากับ b

ตัวอย่างที่ 3.5 โปรแกรมรับข้อมูลตัวเลขจำนวนเต็ม 2 จำนวน หากตัวแปร x มากกว่าตัวแปร y ให้แสดงข้อความ X more than Y แต่หากไม่ใช่ให้แสดงข้อความ X not more than Y

```
#include <stdio.h>
void main()
{
  int x,y;
  printf(" Please Enter X=");
  scanf("%d",&x );
  printf("\n Please Enter Y=");
  scanf("%d",&y);
  (x>y)?printf("\nX more than Y") : printf(" \n X not more than Y");
}
```

ผลการทำงานของโปรแกรมเป็นดังนี้

```
Please Enter X = 30
Please Enter Y = 25
X more than Y
```

ภาพที่ 3.5 ผลลัพธ์บนจอภาพของโปรแกรมตัวอย่างที่ 3.5

แบบฝึกหัดท้ายหน่วยเรียนที่ 3

1) จงแปลงสมการต่อไปนี้ให้เป็นนิพจน์คณิตศาสตร์

$$y = 2x^2 + x \times 3$$

$$y = \frac{2x^2 + 4}{2}$$

2) จงเขียนนิพจน์ตามโจทย์ระบุ

- มีตัวแปร a b และ c ให้เขียนนิพจน์เพื่อหาค่าของ a ซึ่งเกิดจาก b ยกกำลังสองบวกกับ c
- มีตัวแปร a b และ c ให้เขียนนิพจน์เพื่อหาค่าของ a ซึ่งเกิดจาก a คูณกับ c แล้วหารด้วย b
- มีตัวแปร a b และ c ให้เขียนนิพจน์เพื่อเปรียบเทียบว่า a บวก b มีค่ามากกว่า a บวกด้วย c หรือไม่
- มีตัวแปร x เก็บอายุของนักศึกษา ให้ตรวจสอบว่า x มีอายุในช่วง 19 ปี แต่ไม่เกิน 25 ปีหรือไม่
- มีตัวแปร x เก็บตัวเลขจำนวนจริง ให้ตรวจสอบว่า x มีค่าน้อยกว่า 100 หรือมากกว่า 500

3) ให้หาคำตอบของแต่ละนิพจน์ในแต่ละบรรทัด เมื่อกำหนดให้

```
int x =5, y=10, z=15, i;
i = x + 5*y;          //  i = _____
i = x*3 – y;          //  i = _____
i = y/x*z;            //  i = _____
```

4) ให้หาคำตอบของแต่ละนิพจน์ในแต่ละบรรทัด เมื่อกำหนดให้

```
int a =5, b;
float x =12.5,y;
b = 15.5  ;           //  b = _____
y = 10    ;           //  y = _____
b = a/2;              //  b = _____
y = x/2;              //  y = _____
b = a%3*2.5;          //  b = _____
y = x+6/3;            //  y = _____
```

ใบงานที่ 3
ตัวดำเนินการ

จุดประสงค์

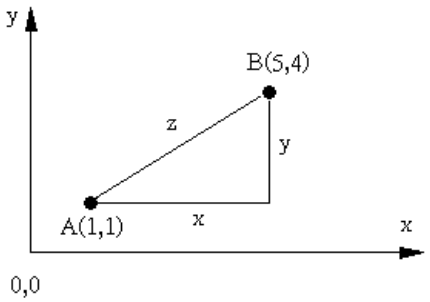
1) ทดลองใช้ตัวดำเนินการแบบต่างๆของภาษาซี
2) ทดลองแปลงสมการทางคณิตศาสตร์เป็นภาษาซี
3) ทดลองเขียนโปรแกรมคำนวณ

จากทฤษฎีที่กล่าวมาในบทที่ 3 จะเห็นว่ามีตัวดำเนินการต่างๆหลายรูปแบบ เช่นตัวดำเนินการกำหนดค่า ตัวดำเนินการทางคณิตศาสตร์ ตัวดำเนินการเปรียบเทียบ หรือตัวดำเนินการเชิงตรรกะ ล้วนแล้วแต่เป็นตัวดำเนินการที่มีความจำเป็นในการเขียนโปรแกรมทั้งสิ้น ทั้งนี้การเขียนโปรแกรมให้ได้ผลลัพธ์ดังที่ผู้ใช้ต้องการ อาจจะต้องใช้เทคนิคและประยุกต์ เพื่อเลือกใช้ตัวดำเนินการต่างๆ เพื่อดำเนินการให้ได้ผลลัพธ์ตามต้องการ ซึ่งส่วนใหญ่แล้วจะเกี่ยวข้องกับการสมการคำนวณทางคณิตศาสตร์ และการคำนวณแบบดิจิตอล หรือตรรกะ ที่นักศึกษาต้องมีพื้นฐานดังกล่าวจึงจะสามารถเข้าใจผลลัพธ์และการทำงานของโปรแกรมได้ แล้วจึงจะสามารถออกแบบโปรแกรมให้สามารถคำนวณหาค่าอื่นๆ ได้ตามต้องการ

เมื่อนักศึกษาเข้าใจการใช้นิพจน์และตัวดำเนินการแล้ว สามารถที่จะประยุกต์ไปใช้กับวิชาอื่นๆที่เกี่ยวกับการคำนวณหาค่าต่างๆ ที่มีขั้นตอนการคำนวณต่างๆ ด้วยเครื่องคิดเลข มาเป็นการเขียนโปรแกรมหาค่า จะทำให้ได้ผลลัพธ์ได้รวดเร็วขึ้น ไม่ต้องเสียเวลาคำนวณทีละขั้นตอน เพียงแต่เขียนโปรแกรมรับค่าอินพุตที่จำเป็น แล้วสร้างสมการคำนวณ และหาผลลัพธ์ต่างๆที่ต้องการ จะเป็นผลให้การคำนวณหาค่าต่างๆนั้นรวดเร็ว และมีความสะดวกมากยิ่งขึ้น

การทดลองที่ 3.1 การคำนวณหาพื้นที่ของรูป 3 เหลี่ยม

การทดลองนี้จะเป็นการคำนวณหาค่าต่างๆ เช่นความยาวด้าน x และ y ความสูงด้าน y และคำนวณหาพื้นที่ของรูปสามเหลี่ยมที่อยู่บนกราฟโคออดิเนต x,y ดังในภาพที่ 3.6 โดยที่รู้ค่าจุดโคออดิเนตเพียง 2 จุด คือจุด A(1,1) และจุด B(4,5) ซึ่งมีลำดับขั้นตอนการทดลองดังนี้



ภาพที่ 3.6 จุดสามเหลี่ยมที่อยู่บนกราฟโคออดิเนต x,y

1) เปิดโปรแกรมคอมไพเลอร์ Turbo C และ สร้างไฟล์ใหม่

2) เขียนโปรแกรมลงในอีดิตเตอร์ตามตัวอย่างโปรแกรมดังนี้

```
#include <stdio.h>
#include<conio.h>
#include <math.h>
void main(void)
{
  int Ax,Ay, Bx, By;
  int x,y;
  float z,Area;
  clrscr();
  Ax = 1;
  Ay =1;
  Bx = 5;
  By = 4;
  x = Bx-Ax;
```

```
    y = By- Ay;
    z = sqrt(x*x + y*y);
    Area = (x*y)/2 ;
    printf(" ***********************************\n");
    printf(" *This is calculate Triangle parameter    *\n");
    printf(" ***********************************\n");
    printf(" Side x =%d\n",x);
    printf(" Side y =%d\n",y);
    printf(" Side z =%.2f\n",z);
    printf(" Area =%.2f\n",Area);
    getch();
}
```

3) บันทึกไฟล์ในเป็นไฟล์ program3_1.c

4) เลือกคำสั่ง Compile และให้บันทึกผล

5) เลือกคำสั่ง RUN และให้บันทึกผลการทดลอง

……

……

……

……

……

การทดลองที่ 3.2 การแปลงค่าองศาเซลเซียสเป็นองศาฟาเรนไฮต์

การทดลองนี้เป็นการเขียนโปรแกรมคำนวณแปลงค่าองศาฟาเรนไฮต์เป็นองศาเซลเซียส แล้วแสดงผลบนจอภาพ ซึ่งมีลำดับขั้นตอนการทดลองดังนี้

1) เปิดโปรแกรมคอมไพเลอร์ Turbo C และ สร้างไฟล์ใหม่

2) เขียนโปรแกรมลงในอีดิตเตอร์ตามตัวอย่างโปรแกรมดังนี้

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int  C=60;
  float F;
   clrscr();
   printf(" ********************************\n");
   printf("*    Calculate  temp   C ==>  F              *\n");
   printf("********************************\n");
   printf("\n");
   printf("\n");
   F = 1.8*C + 32;
   printf("\nThe Fahrenheit temperature  =%.2f  ",F);
   getch();
}
```

3) บันทึกไฟล์ในเป็นไฟล์ program3_2.c

4) เลือกคำสั่ง Compile และให้บันทึกผล

……

……

……

……

……

5) เลือกคำสั่ง RUN และให้บันทึกผลการทดลอง

……
……
……
……
……

6) สรุปผลการทดลอง

……
……
……
……
……
……
……
……
……

งานที่มอบหมาย

1) จงเขียนโปรแกรมแปลงค่าองศาฟาเรนไฮต์เป็นองศาเซลเซียส
2) กำหนดให้ วงจรอนุกรมมีแรงดันไฟฟ้ากระแสตรง 12 โวลท์ ความต้านทาน 120 โอห์ม จงเขียนโปรแกรมและแสดงผลกำลังไฟฟ้าที่ใช้ในวงจร

ใบงานที่ 4
การรับข้อมูลและคำนวณด้วยตัวดำเนินการ

จุดประสงค์

1) ทดลองรับค่าจากคีย์บอร์ดและนำมาใช้ในสมการคำนวณทางคณิตศาสตร์
2) สังเกตุการใช้คำสั่งแสดงผลแบบทศนิยม

เมื่อนักศึกษาเข้าใจการใช้นิพจน์และตัวดำเนินการแล้ว ในใบงานนี้จะเป็นการประยุกต์โดยการนำเข้าข้อมูลจากคีย์บอร์ดด้วยคำสั่ง scanf() แล้วนำข้อมูลเก็บไว้ในตัวแปรที่กำหนด และนำตัวแปรนั้นมาใช้สำหรับการคำนวณทางคณิตศาสตร์ เพื่อคำนวณหาค่าตามสมการ

การทดลองที่ 4.1 การคำนวณหาค่าพื้นที่และเส้นรอบวงของวงกลม

การทดลองนี้จะเป็นการรับข้อมูลรัศมีของวงกลมเพื่อคำนวณหาพื้นที่และเส้นรอบวงของวงกลมขนาดต่างๆ เพื่อให้เข้าใจลักษณะการใช้คำสั่งรับข้อมูลและนำค่าข้อมูลไปคำนวณตามสมการทางคณิตศาสตร์ โดยมีลำดับขั้นตอนการทดลองดังนี้

1) เปิดโปรแกรมคอมไพเลอร์ Turbo C และ สร้างไฟล์ใหม่
2) เขียนโปรแกรมลงในอีดิตเตอร์ตามตัวอย่างโปรแกรมดังนี้

```
#include <stdio.h>
#include<conio.h>
#include <math.h>
void main(void)
{
  float  radius,Area,Circum;
  const pi 3.14;
  clrscr();
```

```
  printf(" **********************************\n");
  printf(" *This is calculate Circle parameter          *\n");
  printf(" **********************************\n");
  printf(" Please enter the radius of circle = ”);
  scanf("%f”,&radius);
  Area = pi*radius*radius ;
  Circum = 2*pi*radius;
  printf("\n Area =%.2f\n",Area);
  printf(" Circum of circle  =%.2f\n",Circum);
  getch();
}
```

3) บันทึกไฟล์ในเป็นไฟล์ program4_1.c

4) เลือกคำสั่ง Compile และให้บันทึกผล

5) เลือกคำสั่ง RUN และให้บันทึกผลการทดลอง

……

……

……

……

……

การทดลองที่ 4.2 การแปลงค่าองศาฟาเรนไฮต์เป็นองศาเซลเซียส

การทดลองนี้เป็นการเขียนโปรแกรมรับค่าองศาฟาเรนไฮต์จากคีย์บอร์ด และจะมีการคำนวณค่าเพื่อแปลงเป็นค่าองศาเซลเซียส แล้วแสดงผลบนจอภาพ ซึ่งมีลำดับขั้นตอนการทดลองดังนี้

1) เปิดโปรแกรมคอมไพเลอร์ Turbo C และ สร้างไฟล์ใหม่

2) เขียนโปรแกรมลงในอีดิตเตอร์ตามตัวอย่างโปรแกรมดังนี้

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int C;
  float F;
   clrscr();
   printf(" **********************************\n");
   printf("*    Translate Temp C ==> F                *\n");
   printf("**********************************\n");
   printf("\n");
   printf("\n");
   printf(" Please Enter Temperature of celsius = ");  scancf("%d",&C);
   F = 1.8*C  +32;
   printf("\nThe Fahrenheit temperature  =%.2f  ",F);
  getch();
}
```

3) บันทึกไฟล์ในเป็นไฟล์ program4_2.c

4) เลือกคำสั่ง Compile และให้บันทึกผล

……

……

……

……

……

5) เลือกคำสั่ง RUN และให้บันทึกผลการทดลอง

……

……

……

……

……

6) สรุปผลการทดลอง

……

……

……

……

……

……

……

……

……

งานที่มอบหมาย

1) จงเขียนโปรแกรมรับค่าองศาฟาเรนไฮต์แล้วคำนวณแปลงค่าและแสดงผลเป็นองศาเซลเซียส
2) จงเขียนโปรแกรมรับค่าแรงดันและโหลดในวงจรอนุกรม คำนวณและแสดงผลค่ากระแส แรงดัน ความต้านทาน และค่ากำลังไฟฟ้าของวงจรไฟฟ้ากระแสตรง