

แผนการสอนประจำบทที่ 2

การรับข้อมูลและการแสดงผล

หัวข้อสำคัญ

1. การแสดงผลข้อมูล
2. การจัดรูปแบบการแสดงผล
3. การรับข้อมูล

วัตถุประสงค์เชิงพฤติกรรม

1. ผู้เรียนสามารถอธิบายคำสั่งที่ใช้ในรับข้อมูลและการแสดงผลข้อมูลแต่ละชนิดได้
2. ผู้เรียนสามารถจัดรูปแบบแสดงผลของข้อมูลแต่ละชนิดได้
3. ผู้เรียนสามารถเขียนโปรแกรมรับข้อมูลและแสดงผลข้อมูลได้

วิธีการสอนและกิจกรรมการเรียนการสอน

1. การบรรยาย
2. การทำแบบฝึกหัด

สื่อที่ใช้ประกอบการสอน

1. เอกสารประกอบการสอน
2. เครื่องคอมพิวเตอร์

การวัดและประเมินผล

1. สังเกตจากความสนใจของผู้เรียน
2. ประเมินจากการตอบคำถามของผู้เรียนและกิจกรรมในชั้นเรียน
3. การทำแบบฝึกหัดท้ายบท

บทที่ 2

การรับข้อมูลและการแสดงผล

2.1 การแสดงผลข้อมูล

คำสั่งที่ใช้ในการแสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ มีทั้งหมด 3 คำสั่ง ได้แก่ printf(), putchar() และ puts()

2.2.1. การแสดงผลข้อมูล printf()

ใช้แสดงผลตัวอักขระหรือข้อความ และต้องระบุรหัสรูปแบบการแสดงผลเพื่อบอกรูปแบบและตำแหน่งในการแสดงผลแต่ละค่า

รูปแบบ
<code>printf("ข้อมูลแสดงผล+การควบคุมการแสดงผล", ตัวแปร);</code>
ตัวอย่าง
<code>printf("Hello");</code> <code>printf("%d", price);</code> <code>printf("the price of computer = %d", price);</code> <code>printf("the score of %s is %f", student,score);</code>

1) รหัสรูปแบบการแสดงผล (Format code)

รหัสรูปแบบการแสดงผลเป็นรหัสที่ใช้กำหนดรูปแบบการแสดงผลข้อมูลจากค่าตัวแปรหรือค่าคงที่ โดยใช้อยู่ในคำสั่ง printf รหัสรูปแบบการแสดงผลแสดงผลดังต่อไปนี้

%d	หมายถึง	พิมพ์ข้อมูลจำนวนเต็มด้วยเลขฐานสิบ
%o	หมายถึง	พิมพ์ข้อมูลจำนวนเต็มด้วยเลขฐานแปด
%x	หมายถึง	พิมพ์ข้อมูลจำนวนเต็มด้วยเลขฐานสิบหก (Lower case)

%X	หมายถึง	พิมพ์ข้อมูลจำนวนเต็มด้วยเลขฐานสิบหก (Upper case)
%u	หมายถึง	พิมพ์ข้อมูลด้วยเลขฐานสิบแบบไม่คิดเครื่องหมาย
%e	หมายถึง	พิมพ์ข้อมูลด้วยตัวเลขแบบวิทยาศาสตร์ เช่น 2.13e45
%f	หมายถึง	พิมพ์ข้อมูลด้วยตัวเลขมีจุดทศนิยม
%c	หมายถึง	พิมพ์ข้อมูลด้วยตัวอักษร 1 ตัว
%s	หมายถึง	พิมพ์ข้อมูลด้วยข้อความ

2) รหัสควบคุม (Control code)

รหัสควบคุมรหัสที่ใช้แทนอักขระพิเศษอื่นและอักขระ White Space

จะใช้เครื่องหมาย\นำหน้าแล้วตามด้วยตัวอักษร, ตัวเลขหรือเครื่องหมายที่ต้องการรหัสควบคุมแสดงผลดังต่อไปนี้

\n	หมายถึง	new line
\t	หมายถึง	horizontal tab
\r	หมายถึง	carriage return
\a	หมายถึง	bell
\0	หมายถึง	null character
\b	หมายถึง	backspace
\'	หมายถึง	'
\?	หมายถึง	?
\"	หมายถึง	"
\\	หมายถึง	\

2.2.2. การแสดงผลอักขระ putchar()

ใช้แสดงผลเฉพาะตัวอักขระ 1 ตัวเท่านั้น

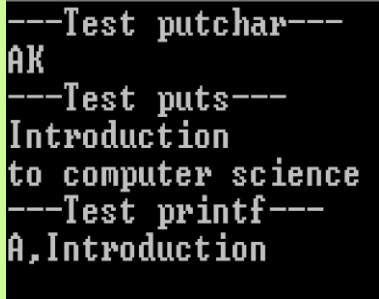
รูปแบบ
<code>putchar(ตัวแปร);</code> หรือ <code>putchar('ตัวอักขระ');</code>
ตัวอย่าง
<pre>char grade = 'A'; putchar(grade); putchar('A');</pre>

2.2.3. การแสดงผลอักขระ puts()

ใช้แสดงผลเฉพาะข้อความเท่านั้น หลังจากแสดงผลข้อความแล้ว
โปรแกรมจะขึ้นบรรทัดใหม่ทันที

รูปแบบ
<code>puts(ตัวแปร);</code> หรือ <code>puts("ตัวอักขระ");</code>
ตัวอย่าง
<pre>char subject[] = "Computer"; puts(subject); puts("Calculus");</pre>

ตัวอย่างของคำสั่ง putchar() และ puts()

โปรแกรม	ผลลัพธ์
<pre> 1. #include<stdio.h> 2. char grade = 'A'; 3. char subject[] = "Introduction"; 4. main() 5. { 6. printf("---Test putchar---\n"); 7. putchar(grade); 8. putchar('K'); 9. printf("\n---Test puts---\n"); 10. puts(subject); 11. puts("to computer science"); 12. printf("---Test printf---\n"); 13. printf("%c,%s",grade,subject); 14. }</pre>	 <pre> ---Test putchar--- AK ---Test puts--- Introduction to computer science ---Test printf--- A,Introduction</pre>

2.2 การจัดรูปแบบการแสดงผล

2.2.1 การจัดรูปแบบข้อความ

%s คือ พิมพ์ข้อมูลด้วยข้อความ

num คือ จำนวนช่องที่จองพื้นที่เพื่อในการแสดงผล

รูปแบบ
printf("% num s",ตัวแปร);
ตัวอย่าง
printf("% 10 s",name);
printf("% -10 s",name);
printf("% .5 s",name);

ถ้ากำหนดให้ `str = "freshy"` และใช้คำสั่ง `printf("%10s",str);` มีการแสดงผลดังภาพ 2.1

				f	r	e	s	h	y
1	2	3	4	5	6	7	8	9	10

ภาพ 2.1 ภาพจำลองการแสดงผลคำสั่ง `printf("%10s",str);`

คำสั่งให้จัดข้อมูลชิดซ้ายจะใส่เครื่องหมายขีด (-) ไว้ข้างหน้า `num` เช่น `printf("%-10s",str);` มีการแสดงผลดังภาพ 2.2

f	r	e	s	h	y				
1	2	3	4	5	6	7	8	9	10

ภาพ 2.2 ภาพจำลองการแสดงผลคำสั่ง `printf("%-10s",str);`

คำสั่งให้ตัดข้อมูลให้แสดงผลเพียงบางส่วนทำได้โดยใส่เครื่องหมายจุด (.) ไว้ข้างหน้า `num` เช่น `printf("%.3s",str);` มีการแสดงผลดังภาพ 2.3

f	r	e
1	2	3

ภาพ 2.3 ภาพจำลองการแสดงผลคำสั่ง `printf("%.3s",str);`

2.2.2 การจัดรูปแบบจำนวนเต็ม

`%d` คือ พิมพ์ข้อมูลด้วยจำนวนเต็มด้วยเลขฐานสิบ

`num` คือ จำนวนช่องที่จองพื้นที่ เพื่อในการแสดงผล

รูปแบบ
<code>printf("%numd",ตัวแปร);</code>
ตัวอย่าง
<code>printf("%10d",age);</code>
<code>printf("%-10d",book);</code>
<code>printf("%9d",number);</code>

ถ้ากำหนดให้ `age = 18` และใช้คำสั่ง `printf("%10d",age);` มีการแสดงผลดังภาพ 2.4

								1	8
1	2	3	4	5	6	7	8	9	10

ภาพ 2.4 ภาพจำลองการแสดงผลคำสั่ง `printf("%10d",age);`

คำสั่ง `printf("%-10d",age);` มีเครื่องหมายลบหน้าจำนวนช่องที่จองพื้นที่ข้อมูลจะจัดชิดซ้าย มีการแสดงผลดังภาพ 2.5

1	8								
1	2	3	4	5	6	7	8	9	10

ภาพ 2.5 ภาพจำลองการแสดงผลคำสั่ง `printf("%-10d",age);`

2.2.3 การจัดรูปแบบจำนวนทศนิยม

`%f` คือ พิมพ์ข้อมูลด้วยจำนวนจริง

`num1` คือ จำนวนช่องที่จองพื้นที่ทั้งหมดของจำนวนทศนิยม

`num2` คือ จำนวนช่องที่จองพื้นที่ของจุดทศนิยม

รูปแบบ
<code>printf("%num1.num2f",ตัวแปร);</code>
ตัวอย่าง
<code>printf("%8.2f",price);</code> <code>printf("%.2f",money);</code> <code>printf("%f",total);</code>

ถ้ากำหนดให้ `price = 18.5` และใช้คำสั่ง `printf("%8.2f, price);` มีการแสดงผลดังภาพ 2.6

			1	8	.	5	0
1	2	3	4	5	6	7	8

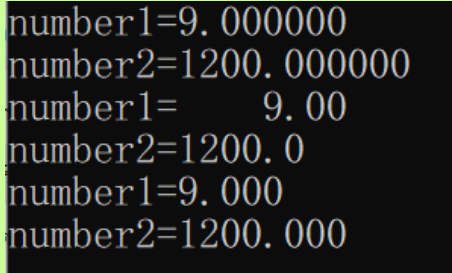
ภาพ 2.6 ภาพจำลองการแสดงผลคำสั่ง `printf("%8.2f, price);`

ใช้คำสั่ง `printf("%-6.1f, price);` มีเครื่องหมายลบหน้าจำนวนช่องที่จองพื้นที่ข้อมูลจะจัดชิดซ้าย มีการแสดงผลดังภาพ 2.7

1	8	.	5		
1	2	3	4	5	6

ภาพ 2.7 ภาพจำลองการแสดงผลคำสั่ง `printf("%-6.1f, price);`

ตัวอย่างการจัดรูปแบบจำนวนทศนิยม

โปรแกรม	ผลลัพธ์
<pre> 1. include <stdio.h> 2. float number1 =9; float number2 =1200; 3. main () 4. { 5. printf("number1=%f \n",number1); 6. printf("number2=%f \n",number2); 7. printf("number1=%8.2f \n",number1); 8. printf("number2=%-6.1f \n",number2); 9. printf("number1=%3f \n",number1); 10. printf("number2=%1.3f \n",number2); 11. }</pre>	 <pre> number1=9.000000 number2=1200.000000 number1= 9.00 number2=1200.0 number1=9.000 number2=1200.000</pre>

2.3 การรับข้อมูล

คำสั่งรับข้อมูลมีทั้งหมด 5 คำสั่ง ได้แก่ scanf(), getchar(), getch(), getche() และ puts()

2.3.1 scanf ()

เป็นคำสั่งที่ใช้รับข้อมูลจากคีย์บอร์ด เช่น ข้อมูลประเภท int, float, char แต่คำสั่ง scanf ต้องระบุรหัสรูปแบบการแสดงผลเพื่อบอกรูปแบบและตำแหน่งในการรับข้อมูลแต่ละค่าเหมือนคำสั่ง printf และสามารถรับข้อมูลได้มากกว่า 1 ข้อมูล ต่อ 1 คำสั่ง หลังจากใส่ข้อมูลครบให้เคาะแป้น Enter เพื่อรับข้อมูล

2.3.1.1 การแสดงผลข้อมูล scanf ()

รูปแบบ
<code>scanf("รหัสควบคุมการแสดงผล", &ตัวแปร);</code>
ตัวอย่าง
<code>scanf("%d",&age);</code>
<code>scanf("%f",&price);</code>
<code>scanf("%f %d", &price,&age);</code>
<code>scanf("%s#%d",&subject,&age);</code>

ตัวอย่างที่ 1 ของคำสั่ง scanf ()

โปรแกรม	ผลลัพธ์
<pre> 1. #include <stdio.h> 2. float grade; 3. main () 4. { 5. printf("Please input your GPA: "); 6. scanf("%f",&grade); //รับค่าตัวแปร grade 7. printf("your GPA = %.2f", grade); 8. }</pre>	<pre> Please input your GPA: 3.25 your GPA = 3.25</pre>

ตัวอย่างที่ 2 ของคำสั่ง scanf ()

โปรแกรม	ผลลัพธ์
<pre> 1. #include <stdio.h> 2. int days,months,years; 3. main () 4. { 5. printf("Please input your Day/Month/Year: "); 6. scanf("%d/%d/%d",&days,&months,&years); //รับค่าตัวแปร days,months,years พร้อมกัน 7. printf("Day is %d, Month is %d, Year is %d ", days,months,years); 8. }</pre>	<pre> Please input your Day/Month/Year: 20/12/2021 Day is 20, Month is 12, Year is 2021</pre>

2.3.2 getchar ()

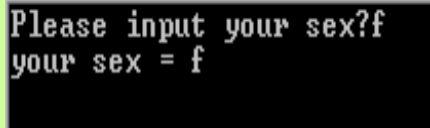
เป็นคำสั่งใช้รับข้อมูลเข้ามาทางแป้นพิมพ์ทีละ 1 ตัวอักษรและรูปแบบคำสั่งต้องกำหนดตัวแปรเพื่อใช้เก็บข้อมูลหลังจากใส่ข้อมูลครบให้**เคาะแป้น Enter** เพื่อรับข้อมูลตัวอักษรที่ป้อน โดยที่ตัวอักษรที่ป้อนจะ**ปรากฏ**ให้เห็นบนหน้าจอภาพด้วย

รูปแบบ
<code>ตัวแปร = getchar();</code>
ตัวอย่าง
<pre>char grade, sex; grade = getchar(); sex = getchar();</pre>

ข้อแนะนำ

โปรแกรมรองรับข้อมูลจากแป้นคีย์บอร์ดที่ผู้ใช้ป้อน จำนวน 1 ตัวอักษรเก็บไว้ในตัวแปร เช่น grade , sex หลังจากนั้นต้องกดแป้น Enterเพื่อให้รับค่าข้อมูล

ตัวอย่างของคำสั่ง getchar ()

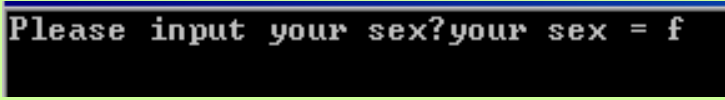
โปรแกรม	ผลลัพธ์
<pre>1. #include <stdio.h> 2. char sex; 3. main () 4. { 5. printf("Please input your sex?"); 6. sex = getchar(); 7. printf("your sex = %c",sex); 9. }</pre>	

2.3.3 getch ()

เป็นคำสั่งใช้รับข้อมูลเข้ามาทางแป้นพิมพ์ทีละ 1 ตัวอักษรและรูปแบบคำสั่งต้องกำหนดตัวแปรเพื่อใช้เก็บข้อมูลเช่นเดียวกับ getchar() หลังจากใส่ข้อมูลครบไม่ต้องเคาะแป้น Enter โดยที่ตัวอักษรที่ป้อนจะไม่ปรากฏให้เห็นบนหน้าจอภาพด้วย

รูปแบบ
<code>ตัวแปร = getch();</code>
ตัวอย่าง
<pre>char grade, sex; grade = getch(); sex = getch();</pre>

ตัวอย่างของคำสั่ง getch ()

โปรแกรม
<pre>1. # include <stdio.h> 2. char sex; 3. main () 4. { 5. printf("Please input your sex?"); 6. sex = getch (); 7. printf("your sex = %c",sex); 8. }</pre>
ผลลัพธ์


2.3.5 getche ()

เป็นคำสั่งใช้รับข้อมูลเข้ามาทางแป้นพิมพ์ทีละ 1 ตัวอักษรเช่นเดียวกับคำสั่ง getchar () และ getch() และรูปแบบคำสั่งต้องกำหนดตัวแปรเพื่อใช้เก็บข้อมูลเช่นเดียวกับ getchar() หลังจากใส่ข้อมูลครบไม่ต้องเคาะเป็น Enter โดยที่ตัวอักษรที่ป้อนจะปรากฏให้เห็นบนหน้าจอภาพด้วย

รูปแบบ
ตัวแปร = getche();
ตัวอย่าง
<pre>char grade, sex; grade = getche(); sex = getche();</pre>

ข้อแตกต่างระหว่าง getchar () getch() และ getche()

getchar () เมื่อใส่ข้อมูลเสร็จต้องเคาะ enter ข้อมูลปรากฏหน้าจอ

getch () ไม่ต้องเคาะ enter ข้อมูลไม่ปรากฏหน้าจอ

getche () ไม่ต้องเคาะ enter ข้อมูลปรากฏหน้าจอ

2.3.4 gets ()

เป็นคำสั่งใช้รับข้อมูลเข้ามาทางแป้นพิมพ์ได้ทั้งอักขระและข้อความ (รับข้อมูลมากกว่า 1 ตัวอักษร) และรูปแบบคำสั่งไม่ต้องกำหนดตัวแปรเพื่อใช้เก็บข้อมูล หลังจากใส่ข้อมูลครบให้เคาะเป็น Enter โดยที่ตัวอักษรที่ป้อนจะปรากฏให้เห็นบนหน้าจอภาพด้วย

รูปแบบ
gets (ตัวแปร);
ตัวอย่าง
<pre>char subject[10]; gets(subject); char student_code[10]; gets(student_code);</pre>

ข้อแนะนำ

โปรแกรมจะจองพื้นที่ของตัวแปรชื่อ subject ซึ่งเป็นอักขระ ไว้ 10 ตัว โดยตัวแปร subject จะเก็บข้อมูลได้ยาวไม่เกิน 9 ตัวอักษร เพื่อให้ subject ตัวที่ 10 (ตัวสุดท้าย) เก็บ \0 เอาไว้

ตัวอย่างที่ของคำสั่ง gets ()

โปรแกรม	ผลลัพธ์
<pre> 1#include <stdio.h> 2. char subject[10]; 3. main () 4. { 5. printf("Please input your subject? "); 6. gets(subject); 7. printf("your subject = %s",subject); 9. }</pre>	 <pre> Please input your subject? CALCULUS your subject = CALCULUS</pre>

คำถามท้ายบทที่ 2

1. จงประมวลผลโปรแกรมต่อไปนี้พร้อมทั้งแสดงผลลัพธ์ของโปรแกรมที่ปรากฏบนหน้าจอคอมพิวเตอร์ และแสดงตำแหน่งสุดท้ายของ Cursor และกำหนดให้

- แสดงช่องว่างโดยใช้เครื่องหมาย □
- แสดงตำแหน่งสุดท้ายของ cursor โดยใช้เครื่องหมาย _

โปรแกรม	ผลลัพธ์ (แสดงผลลัพธ์ตามบรรทัดบนหน้าจอคอมพิวเตอร์)
1. #include<stdio.h>	
2. char grade = 'A';	1.
3. char class[] = "C Programming";	2.
4. int num1 = 10;	3.
5. float num2 = 55;	4.
6. main()	5.
7. {	6.
8. putchar(grade);	7.
9. puts(class);	8.
10. printf("class=%15s\n",class);	9.
11. num1 = num1*5;	10.
12. num2 = num1;	11.
13. printf("num1=%d,num2=%f\n",num1,num2);	12.
14. printf("num1=%-5d,num2=%1.2f\n",num1,num2);	13.
15. num1*=5;	14.
16. printf("num1=%d\n",num1+5);	15.
17. --num1;	
18. printf("num1=%d\n",num1);	
19. }	

2. จงเขียนโปรแกรมจัดรูปแบบข้อความให้แสดงผลดังต่อไปนี้ เมื่อกำหนดให้มีการประกาศตัวแปรด้วยคำสั่ง `char subject[] = "Turbo C";` และ `□` หมายถึงช่องว่าง

```

□□□□□Turbo□C
Turbo□C□□□
Turbo
Turbo□
Turbo□C

```

3. จงเขียนโปรแกรมจัดรูปแบบเลขจำนวนเต็มให้แสดงผลดังต่อไปนี้ เมื่อกำหนดให้มีการประกาศตัวแปรด้วยคำสั่ง `int number1 = 10, number2 = 1500;` และ `□` หมายถึงช่องว่าง

```

Number1 = 10
Number2 = 1500
Number1 = 10□□□
Number2 = 1500□□□
Numbe2 = □□1500

```

4. จงเขียนโปรแกรมจัดรูปแบบเลขจำนวนจริงให้แสดงผลดังต่อไปนี้ เมื่อกำหนดให้มีการประกาศตัวแปรด้วยคำสั่ง `float value1 = -1.25, value2 = 15.12345;` และ `□` หมายถึงช่องว่าง

```

Value1 = -1.25□□□□
Value2 = 15.12□□
Value1 = -1.25
Value2 = □□□15.12
Value2 = 15.12

```

5. จงเขียนโปรแกรมรับข้อมูลส่วนตัวของนิสิต ได้แก่ รหัสนิสิต ชื่อนามสกุลนิสิต รหัสสาขาวิชา อายุ และเกรดเฉลี่ย ผ่านแป้นคีย์บอร์ดและกำหนดให้แสดงผลบนหน้าจอด้วยตัวอย่างด้านล่าง

```
My name is Jirawan Charosensuk and student id is 6430200211  
I am 19 year old year old and study S05  
My GPA is 3.25
```

6. จงบอกรหัสรูปแบบการแสดงผลของชนิดข้อมูลจำนวนเต็มที่ใช้กับคำสั่ง printf
7. จงอธิบายความหมายของอักขระ White Space พร้อมยกตัวอย่างประกอบ
8. จงอธิบายความแตกต่างระหว่างคำสั่ง getchar() และ gets()
9. จงอธิบายความแตกต่างระหว่างคำสั่ง printf() และ puts()
10. จงอธิบายความแตกต่างระหว่างคำสั่ง scanf() และ gets()