INPUT AND MATH IN C

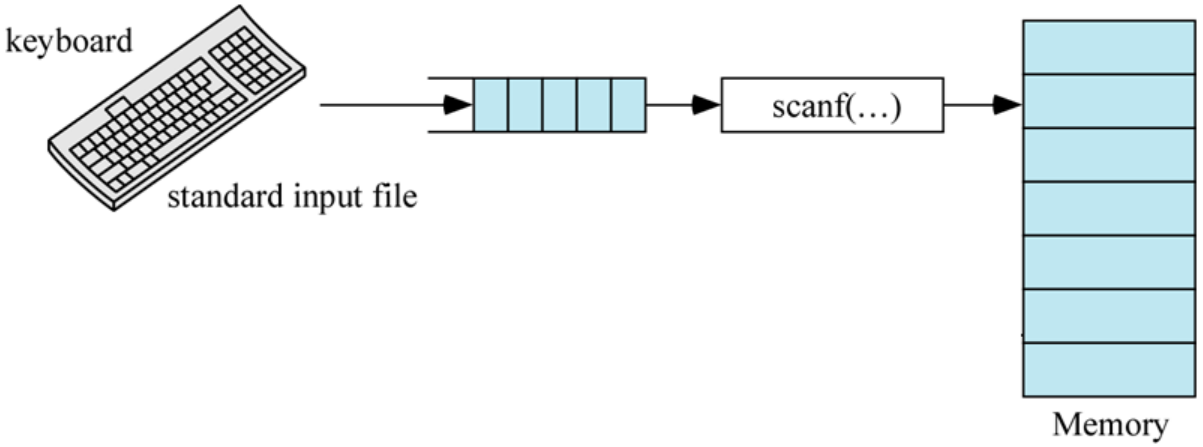
030523300- Computer Programming
Asst. Prof. Dr. Choopan Rattanapoka

การป้อนข้อมูลทางแป้นพิมพ์

- การป้อนข้อมูลทางแป้นพิมพ์เป็นวิธีหนึ่งที่ใช้ในการกำหนดค่าให้กับตัวแปรในโปรแกรมต้นฉบับ
- ซึ่งการทำงานนั้นก็คือการนำค่าที่ได้รับจากแป้นพิมพ์ไปเก็บไว้ในหน่วยความจำโดยตำแหน่งของหน่วยความจำนั้นก็คือตำแหน่งที่ตัวแปรใช้ในการเก็บข้อมูล
- ในภาษาซีมีฟังก์ชันสำหรับป้อนข้อมูลที่นิยมใช้คือ
  - scanf ( )
  - getchar ( )
  - getche ( )
  - getch ()

ฟังก์ชัน scanf( )

- เมื่อโปรแกรมทำงานถึงฟังก์ชันนี้จะหยุดรอเพื่อให้ผู้ใช้ป้อนข้อมูลโดยข้อมูลที่ป้อนจะแสดงบนจอภาพ เมื่อป้อนข้อมูลเสร็จกด **Enter** ข้อมูลจะผ่านตัวฟังก์ชัน **scanf** เพื่อแปลงค่าที่ได้ไปเก็บไว้ยังหน่วยความจำหลัก



การใช้งาน scanf

scanf (“format code” , &var);

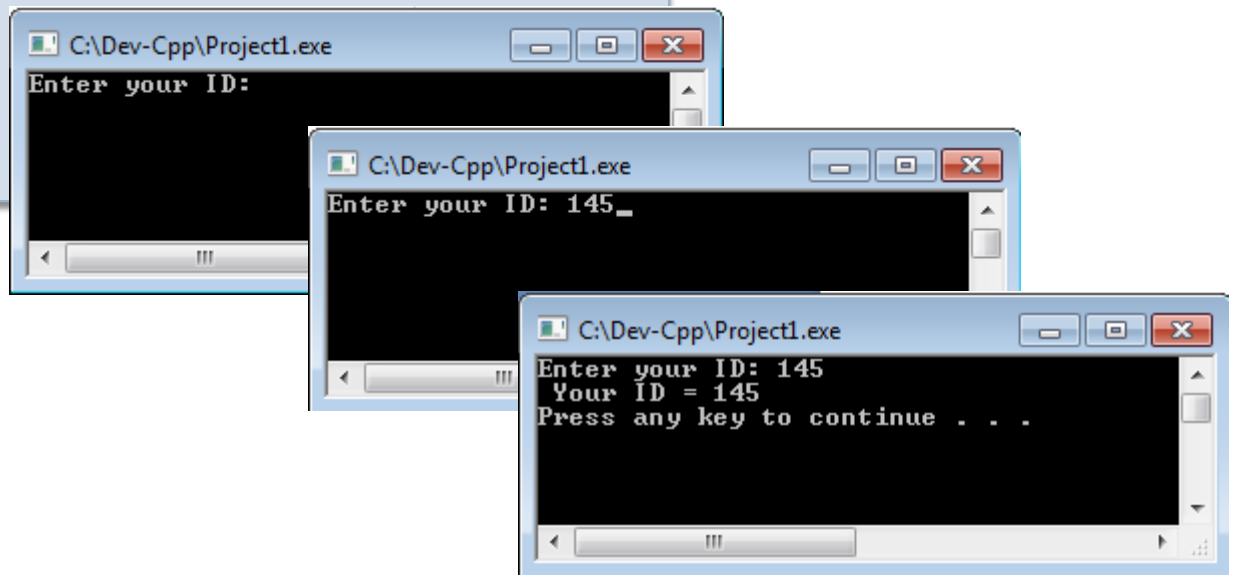
- Format code จะมีลักษณะการใช้งานคล้ายกับ format code ที่ใช้กับฟังก์ชัน printf ซึ่งเป็นตัวกำหนดประเภทของข้อมูลที่ต้องการรับผ่านแป้นพิมพ์
- var คือชื่อตัวแปรที่ต้องการใช้ในการเก็บข้อมูลที่รับจากแป้นพิมพ์

Format code สำหรับ scanf

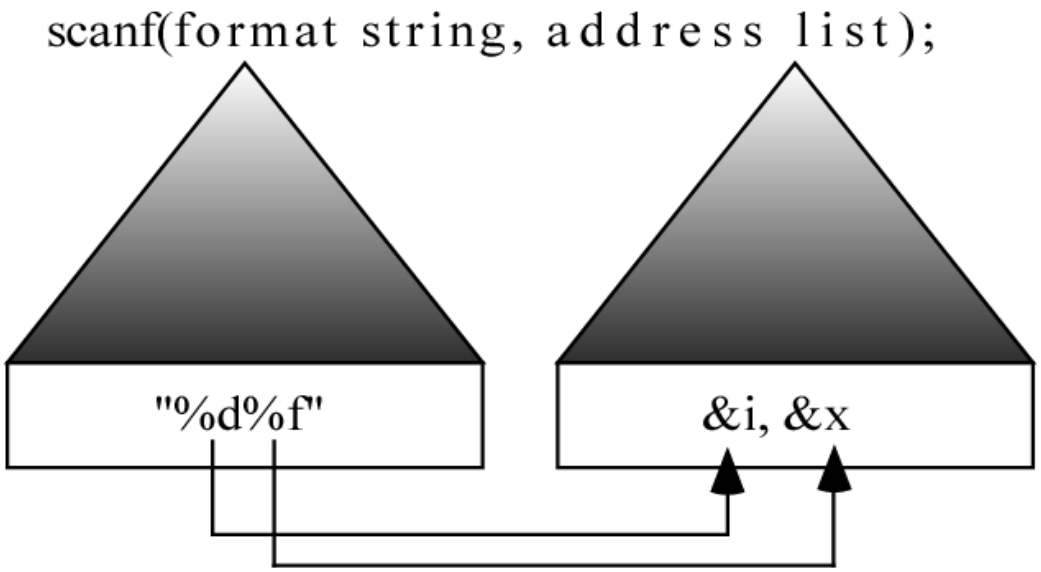
รหัสรูปแบบ	ความหมาย
%c	ใช้กับตัวแปรที่เก็บค่าเป็นตัวอักษรเพียงตัวเดียว
%s	ใช้กับตัวแปรที่เก็บค่าเป็นข้อความที่เก็บในตัวแปรชุด
%d	ใช้กับตัวแปรที่เก็บค่าเป็นเลขจำนวนเต็ม
%u	ใช้กับตัวแปรที่เก็บค่าเป็นเลขจำนวนเต็มบวก
%f	ใช้กับตัวแปรที่เก็บค่าที่เป็นเลขทศนิยม (float)
%lf	ใช้กับตัวแปรที่เก็บค่าที่เป็นเลขทศนิยม (double)

ตัวอย่างโปรแกรม 1

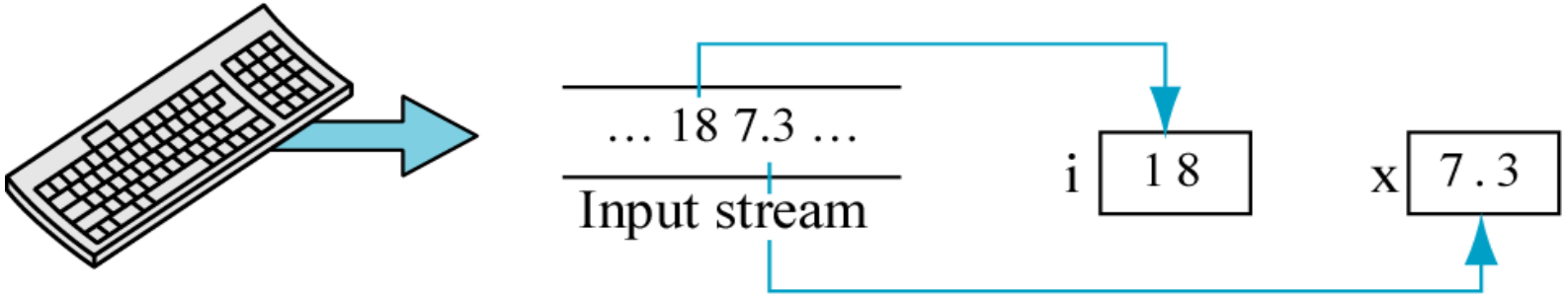
```
#include<stdio.h>
int main(int argc, char **argv) {
    int   x;   // กำหนดตัวแปร x ที่เป็นชนิด integer
    printf("Enter your ID: ");
    scanf("%d", &x);          // รับข้อมูลเก็บไว้ใน x
    printf(" Your ID = %d\n", x);    // พิมพ์จำนวนเต็ม x

    return 0;
}
```



รูปแบบการใช้ scanf เก็บค่ามากกว่าหนึ่งตัวแปร

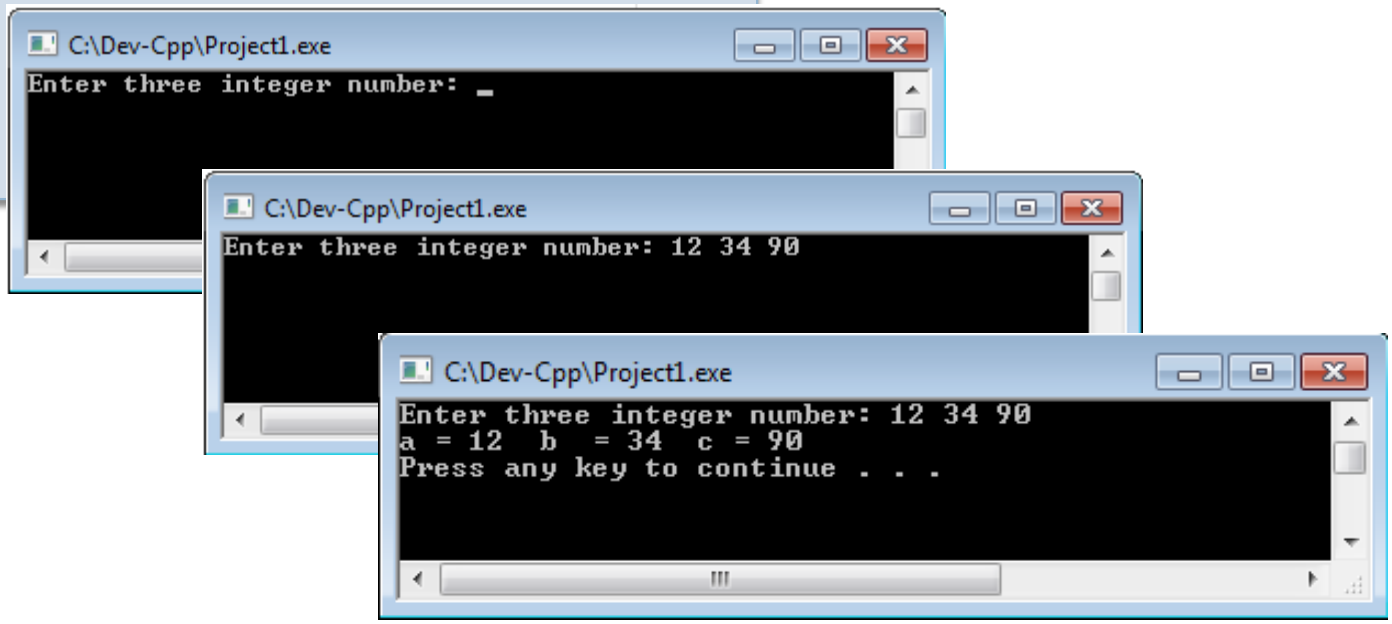


scanf("%d%f", &i, &x);



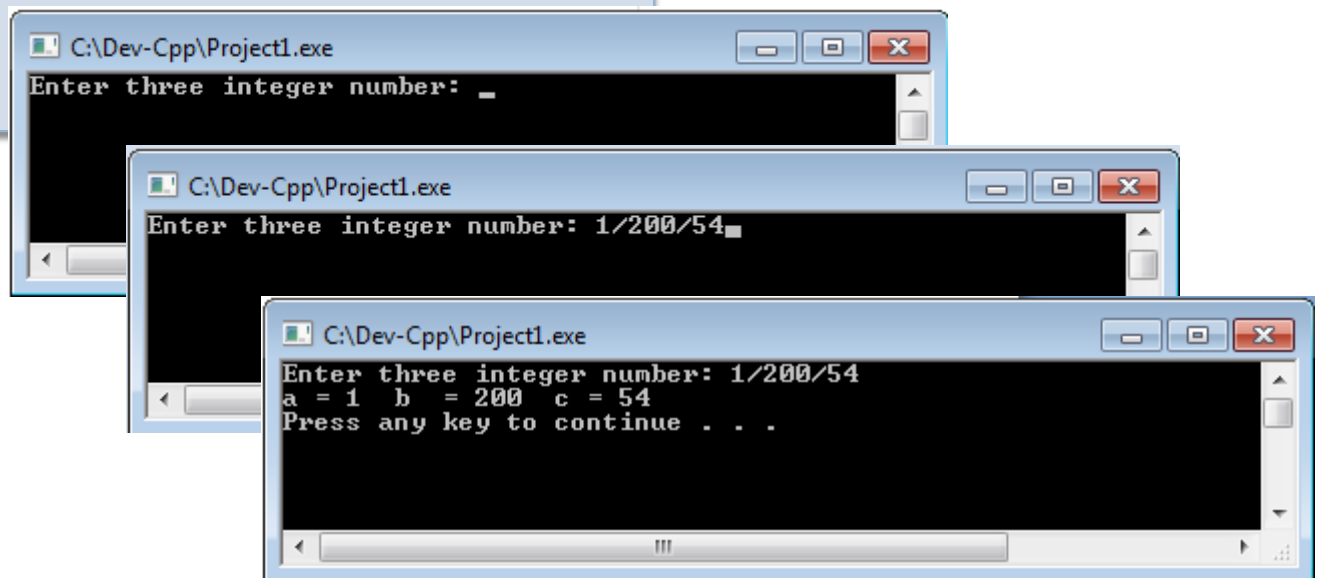
ตัวอย่างโปรแกรม 2

```
#include<stdio.h>
int main(int argc, char **argv) {
    int   a,  b,  c;
    printf("Enter three integer number: ");
    scanf("%d %d %d", &a, &b, &c);
    printf("a = %d  b  = %d  c = %d\n", a, b, c);
}
```



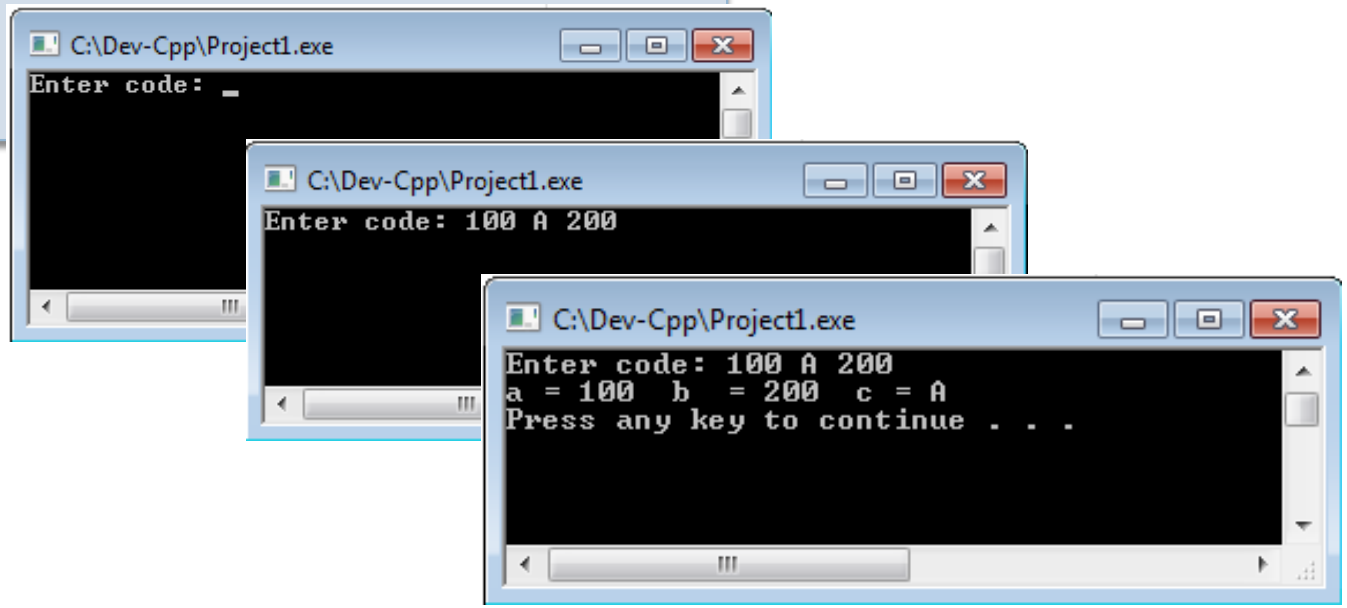
ตัวอย่างโปรแกรม 3

```
#include<stdio.h>
int main(int argc, char **argv) {
    int   a,  b,  c;
    printf("Enter three integer number: ");
    scanf("%d/%d/%d", &a, &b, &c);

    printf("a = %d  b  = %d  c = %d\n", a, b, c);
}
```



ตัวอย่างโปรแกรม 4

```
#include<stdio.h>
int main(int argc, char **argv) {
    int   a,  b;
    char  c;
    printf("Enter code: ");
    scanf("%d %c %d", &a, &c, &b);
    printf("a = %d  b  = %d  c = %c\n", a, b, c);
}
```



ฟังก์ชัน getchar( ), getche(), getch()

- ทั้ง 3 ฟังก์ชันนี้จะเป็นการรับค่าตัวอักษรเพียง 1 ตัวจากทางแป้นพิมพ์
- getchar()
  - รูปแบบการใช้งาน ch = getchar( );
  - แสดงตัวอักษรที่ป้อนออกทางหน้าจอ ผู้ใช้ต้องกด Enter หลังจากป้อนข้อมูล
- getche()
  - รูปแบบการใช้งาน ch = getche();
  - แสดงตัวอักษรออกทางหน้าจอ ไม่ต้องกด Enter หลังป้อนข้อมูล
- getch()
  - รูปแบบการใช้งาน ch = getch();
  - ไม่แสดงตัวอักษรออกทางหน้าจอ ไม่ต้องกด Enter หลังป้อนข้อมูล

คณิตศาสตร์ในภาษาซี

- ตัวดำเนินการทางคณิตศาสตร์พื้นฐานในภาษาซี คือ
  - \+ บวก
  - \- ลบ
  - * คูณ
  - / หาร(ถ้าเป็นจำนวนเต็ม จะไม่คิดเศษ)
  - % หารเอาแต่เศษ (ใช้ได้เฉพาะกับจำนวนเต็ม)

แบบฝึกหัด 1

- จงหาค่าของตัวแปร i, j, k ตามคำสั่งต่อไปนี้ตามลำดับ

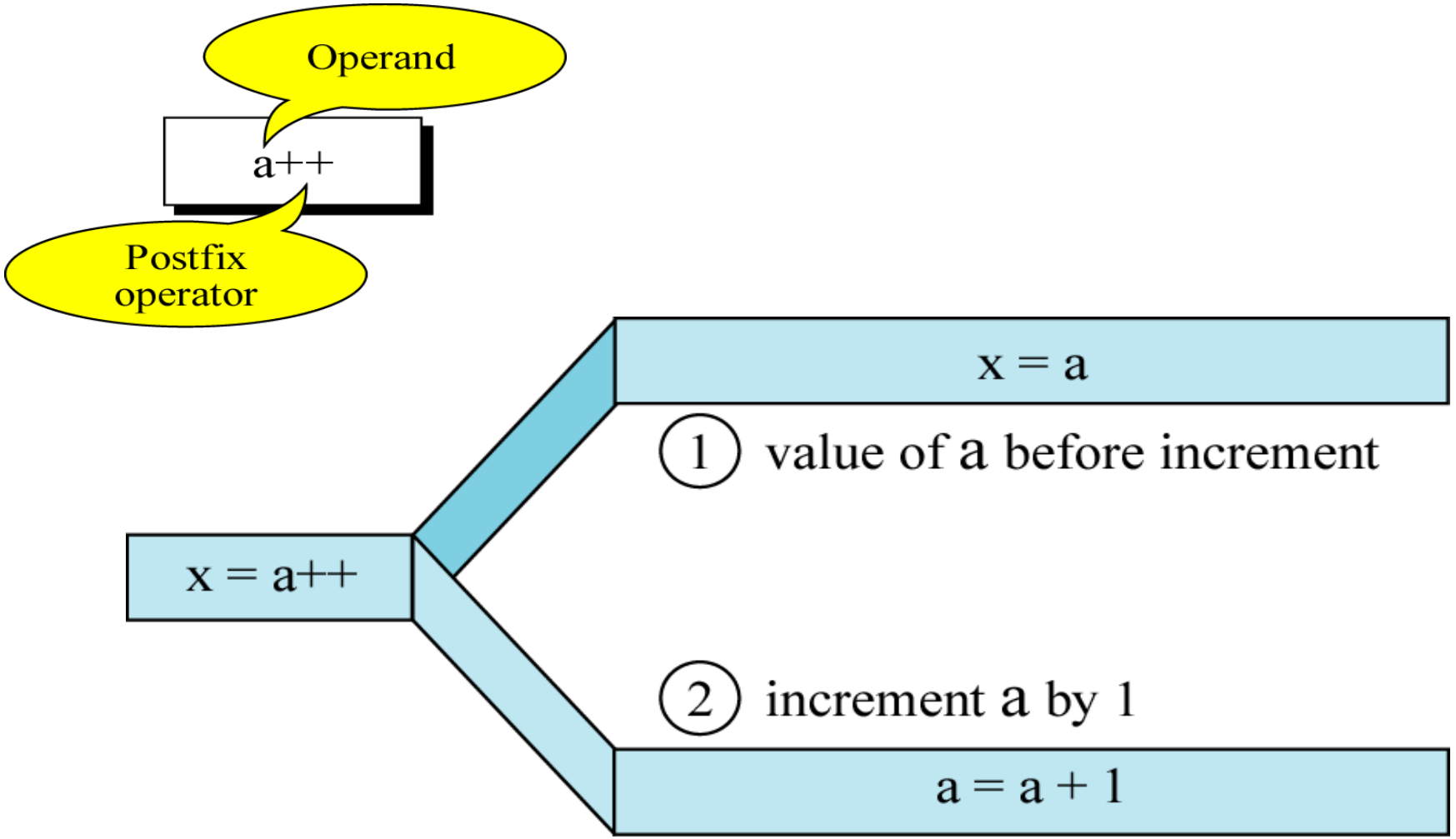
	i	j	k
int i = 1, j = 2 , k;	1	2	?
k = i + j;	1	2	3
i = i + (k * j);	7	2	3
j = i / 2;	7	3	3
k = i % 2;	7	3	1
i = (j + k) * 3;	12	3	1

แบบฝึกหัด 2

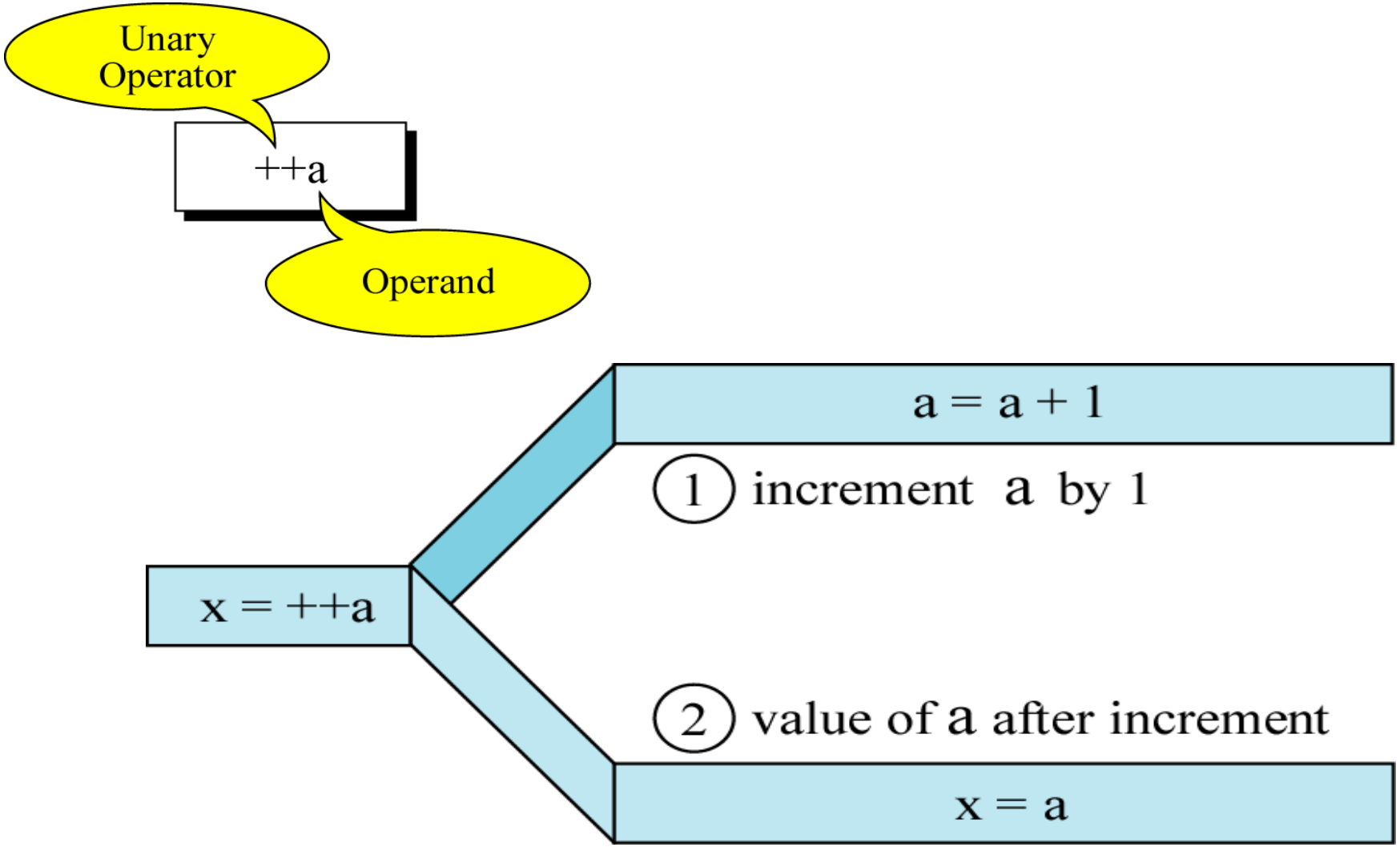
- จงหาค่าของตัวแปร i, j, k ตามคำสั่งต่อไปนี้ตามลำดับ

	x	y
double x=1.0, y=2.0;	1.0	2.0
x = y + 5.0;	7.0	2.0
y = x / 2.0;	7.0	3.5
y = (x * 3.0) + 4.0;	7.0	25.0
x = -0.5 - y;	-25.5	25.0

เครื่องหมาย ++ (1)


Operand
a++
Postfix operator
x = a
1 value of a before increment
x = a++
2 increment a by 1
a = a + 1

เครื่องหมาย ++ (2)


Unary Operator
++a
Operand
a = a + 1
1 increment a by 1
x = ++a
2 value of a after increment
x = a

เครื่องหมาย -- และตัวอย่าง

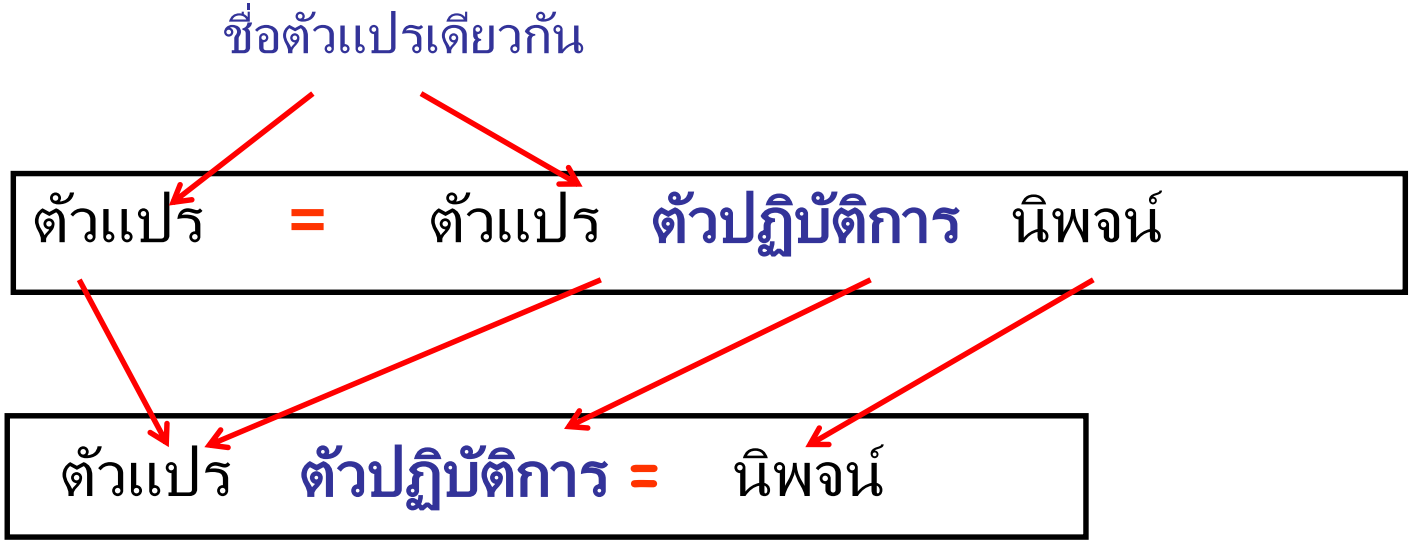
- **เครื่องหมาย --**
  - ทำงานเหมือนกับเครื่องหมาย ++ แต่เป็นการลดค่าตัวแปรลง 1 ค่า
  - X -- ; → X = X – 1;

ตัวอย่าง

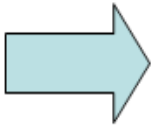
```
int i=1, j=2, k;
i++;
--j;
j = i++;
k = ++j  + i -- ;
k = j++  + --i;
```

i	j	k
1	2	?
2	2	?
2	1	?
3	2	?
2	3	6
1	4	4

การเขียนนิพจน์ในรูปย่อ




```
x = x - 5.0;
x = 3.0 * x;
x = x % ( y + z * a );
x = x / ( 2.0 * x );
```



```
x -= 5.0;
x *= 3.0;
x %= ( y + z * a );
x /= ( 2.0 * x );
```

แบบฝึกหัด 3

	x	y
`int x=1, y=2;`	1	2
`x  += 9;`	10	2
`x /= y;`	5	2
`y *= (x + 5);`	5	20
`y -= (x % 3);`	5	18

นิพจน์คณิตศาสตร์

- นิพจน์คณิตศาสตร์ประกอบด้วยค่าคงที่ ตัวแปร ฟังก์ชัน หรือการรวมกันของค่าคงที่ ตัวแปรฟังก์ชัน ตัวปฏิบัติการทางคณิตศาสตร์ และเครื่องหมายวงเล็บเปิดปิด
- ลำดับในการประมวลผล
  - วงเล็บคร่อมนิพจน์จะถูกประมวลผลก่อน
  - ถ้ามีวงเล็บหลายชั้น นิพจน์ที่อยู่ชั้นในสุดจะถูกประมวลผลก่อน
  - เครื่องหมายคณิตศาสตร์ที่มีลำดับเดียวกันจะถูกประมวลผลจากซ้ายไปขวา

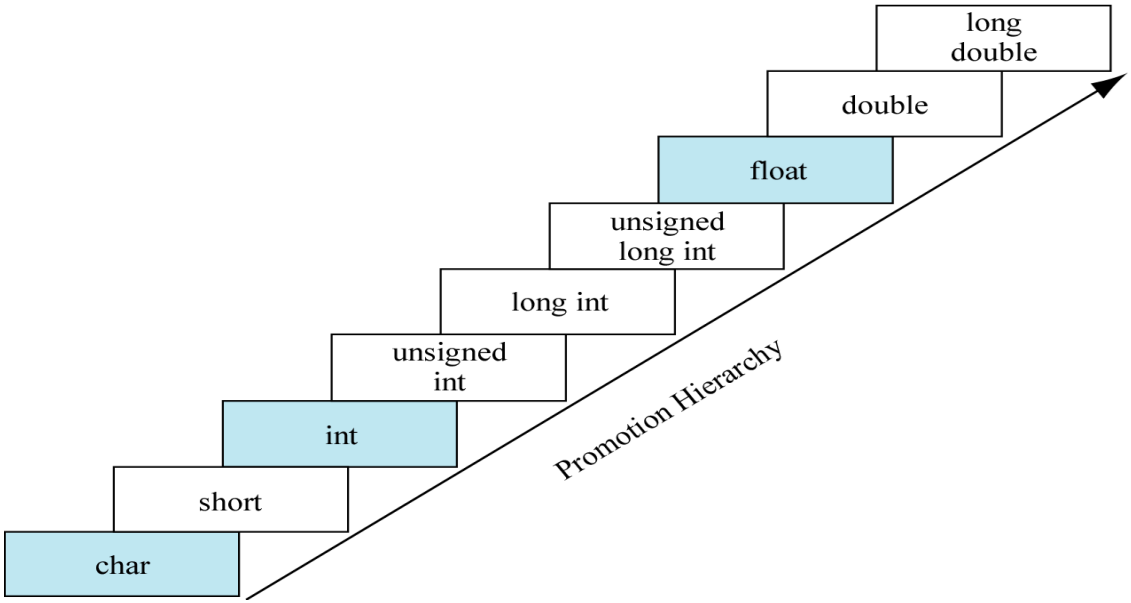
เครื่องหมายคณิตศาสตร์	ลำดับการประมวลผล
++ , --	1
- (เครื่องหมายลบหน้าตัวเลข)	2
* , / , %	3
+ , -	4

แบบฝึกหัด 4

- -2 + 5 * 2
- 10/2 * 3
- 6 / 2 + 3 * (4 % 2)
- (5+2) * 15 % 4
- 6 + 2 * 2 – 6 / 2

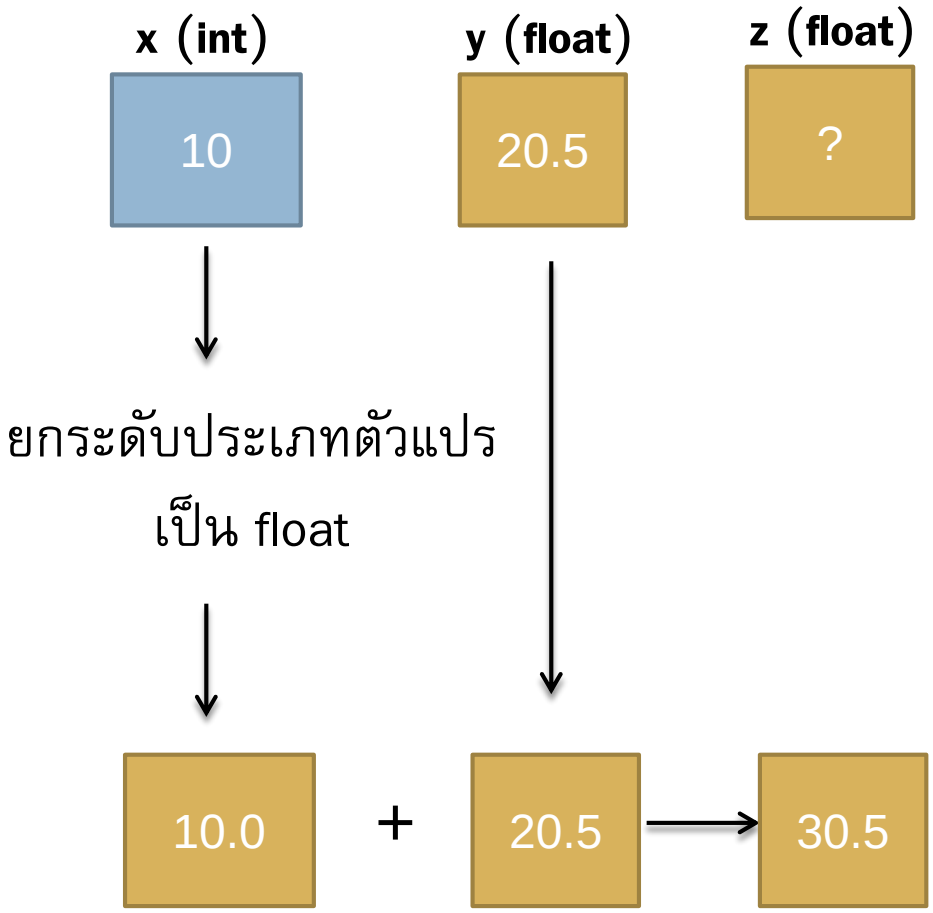
การเปลี่ยนประเภทของข้อมูล

- เมื่อให้ค่ากับตัวแปรที่รับข้อมูลได้ช่วงกว้างกว่า
- + – * / หรือ % ในนิพจน์รับข้อมูลต่างประเภทกัน
- เมื่อมีการคำนวณทางคณิตศาสตร์ ตัวถูกกระทำที่อยู่ในระดับต่ำกว่าจะถูกเปลี่ยนชนิดให้เหมือนกับตัวที่อยู่สูงกว่า ตามลำดับขั้นดังรูป



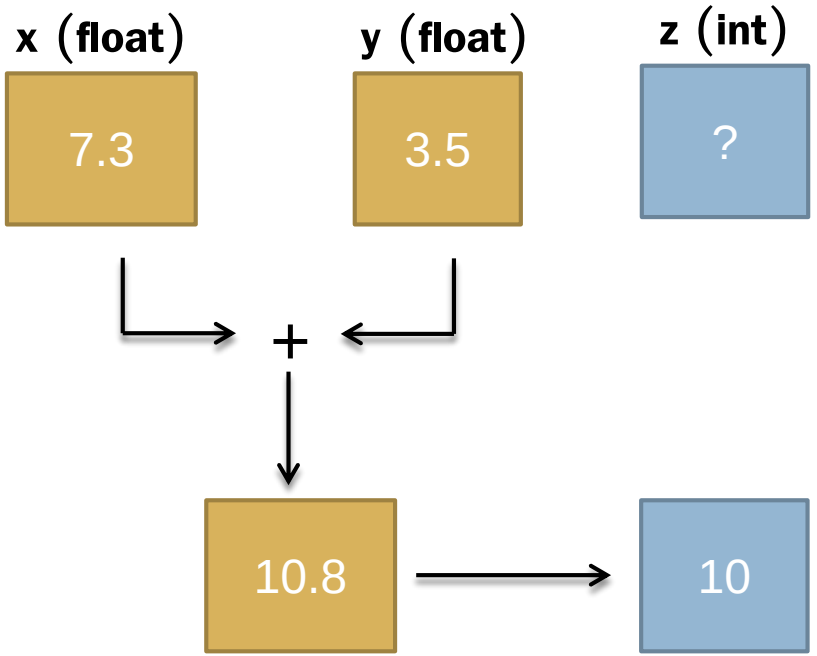
ตัวอย่าง : การเปลี่ยนประเภทของข้อมูล

```
int main(void) {
    int x = 10;
    float y = 20.5, z;
    z = x + y;
}
```



ตัวอย่าง : การเปลี่ยนประเภทของข้อมูล

```
int main(void) {
    float x = 7.3;
    float y = 3.5;
    int z;
    z = x + y;
}
```



ตัวแปล z มีประเภทของข้อมูลเป็น int
ไม่สามารถเก็บตัวเลขที่มีจุดทศนิยมได้ ดังนั้นค่าที่คำนวณได้จะถูกตัดจุดทศยมทิ้ง เพื่อแปลงเป็นจำนวนเต็ม
(ไม่มีการปัดจุด)

การเปลี่ยนประเภทข้อมูลโดยตรง (Cast)

- รูปแบบ

(Data type) expression

- ตัวอย่าง
  - (float)a
  - (float)(x+y)
  - (int)(a/10) มีค่าไม่เท่ากับ (int)a/10

แบบฝึกหัด 5

- จงหาค่าของ a,b,c,d, และ e

```
float a;      a = (int)5.5 + (float)10.4;

int b;        b = (int)5.5 + 10.4;

int c;        c = (int)(5.5/1.1);

int d;        d = (int)5.5/1.1;

int e;        e  = 5.5/(int)1.1;
```