เนื้อหาที่จะเรียนในวันนี้

1. ทบทวน Structure
2. ทบทวน Pointer
3. Pointer to Array
4. Pointer to Structure
5. การใช้งาน pointer เมื่อเขียนโปรแกรม และการ new

บทที่ 2

6. Recursive และการทำงานของ recursive
7. คณิตศาสตร์พื้นฐาน
8. Big O

Structure

1.Definition structure

2.ประกาศตัวแปร

3.การใส่ค่าลงในตัวแปร

- Assign ค่าด้วยเครื่องหมาย =
- รับข้อมูลจาก keyboard
- ใส่ค่าตอนประกาศตัวแปร

4. การดึงค่ามาใช้

3. Structure

type ที่ user สร้างเองโดยประกอบข้อมูลเป็น type

Structure is a user-defined datatype which allows us to combine data of different types together.

```
struct student
{       int Id;
        float Grade;
        char Gender;
};
struct  student  std1;
```

9 bytes

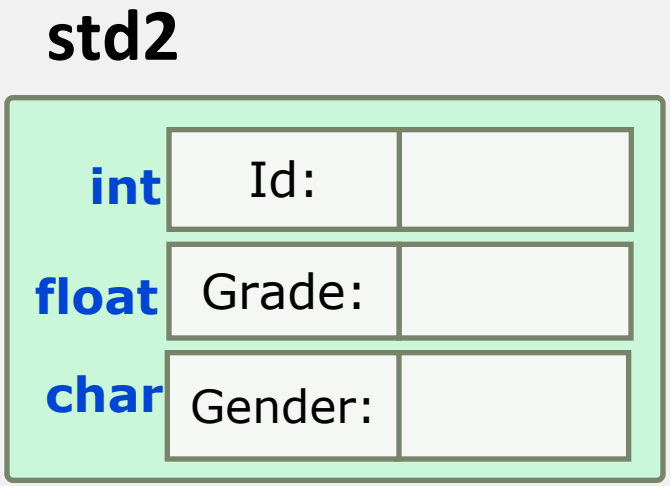
std1

int	Id:	
float	Grade:	
char	Gender:	

การใช้งาน

1.Definition structure ✓
2.ประกาศตัวแปร ✓
3.การใส่ค่าลงในตัวแปร
4.การดึงค่ามาใช้

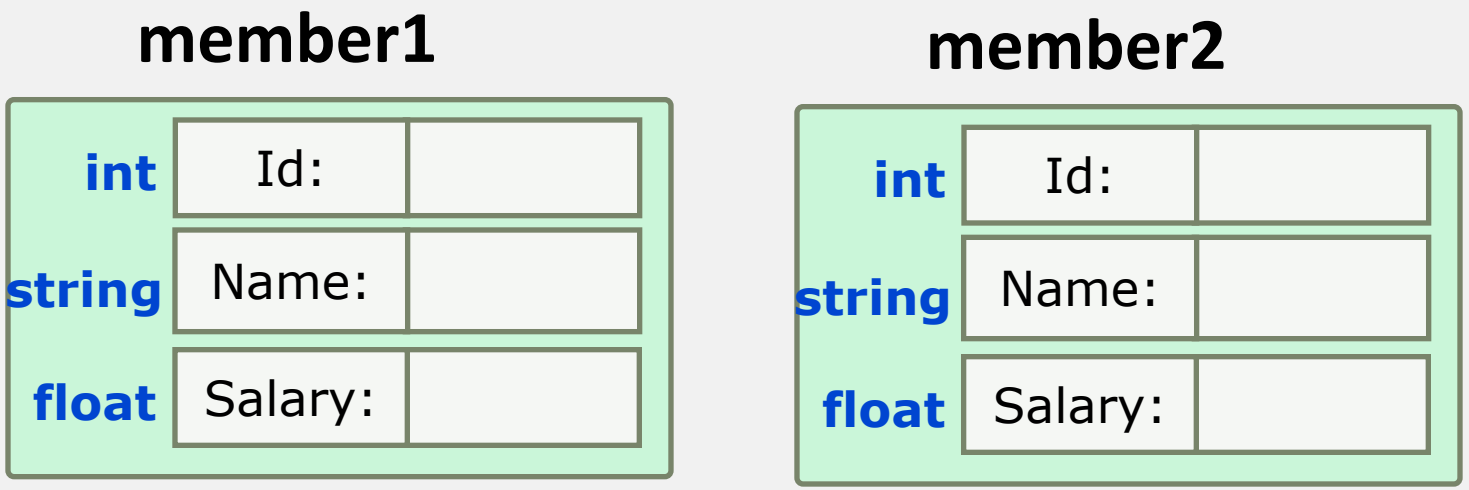
```
struct student
{    int Id;
     float Grade;
     char Gender;
}std2;
```



3.1 การ defined structure และประกาศตัวแปร

```
struct employee
{    int Id;
     string Name;
     float Salary;
};
struct  employee member1, member2;
```

หมายถึงกับข้อมูลใน C++



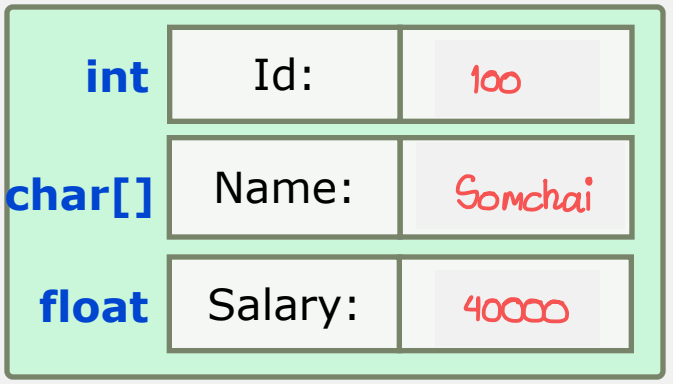
3.2 การใส่ค่าลงในตัวแปร structure

```
member1.Id=100;
member1.Name= "Somchai";
member1.Salary=40000;
```

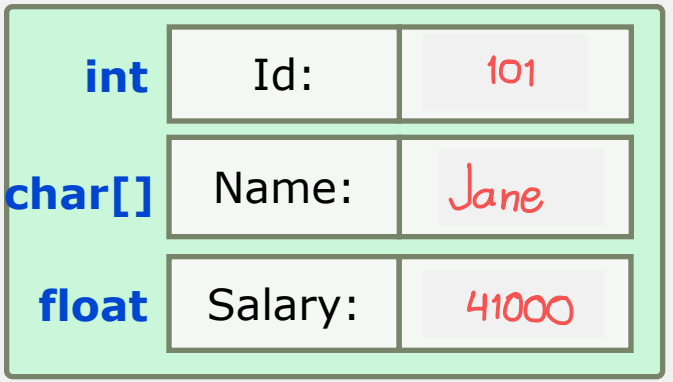
```
cin >> member2.Id;
cin >> member2.Name;
cin >> member2.Salary;
```

101
Jane
41000

member1



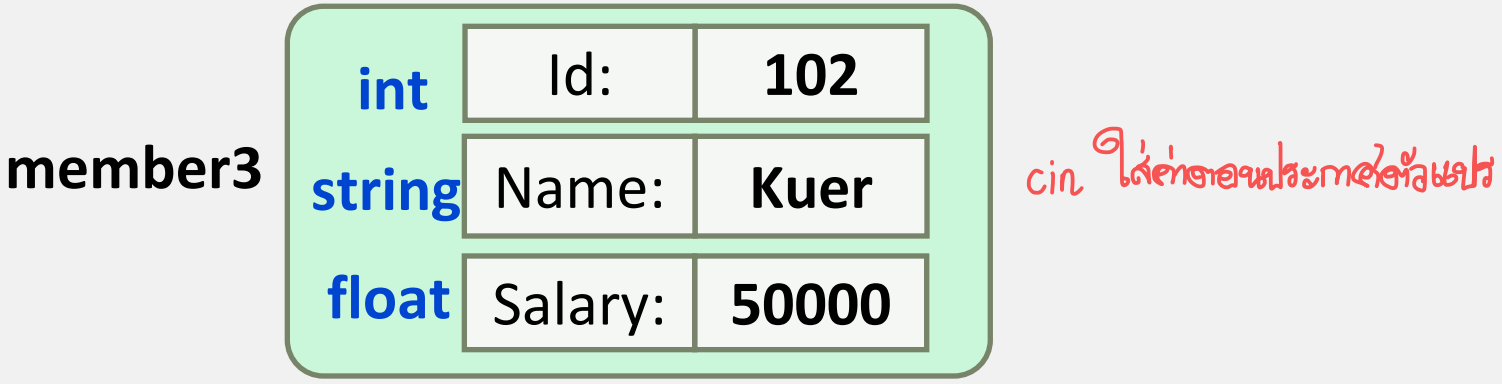
member2



3.3 structure initialization(ใส่ค่าเริ่มต้น)

init = initialization (ประกาศค่า)

```
struct  structname  variable = { val1, val2,  ... };
```



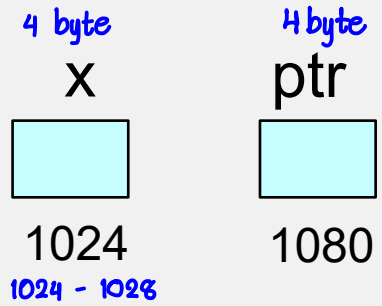
cin ใส่ค่าตอนประกาศตัวแปร

```
struct  employee  member3 = { 102, "Kuer", 50000};
```

4. Pointer

1. Declaration of a pointer

```
int x,  *ptr;
```



pointer to integer
เก็บตำแหน่ง ตำแหน่งนั้นมี integer อยู่

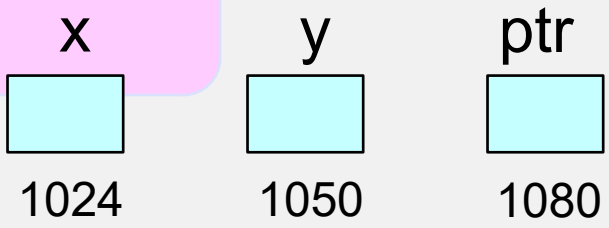
2. Pointer Operator

เครื่องหมาย

& Address Operator

* Indirection Operator

```
int x, y, *ptr;
x=5;
ptr = &x;
y = *ptr;
```



&x = Address = 1024

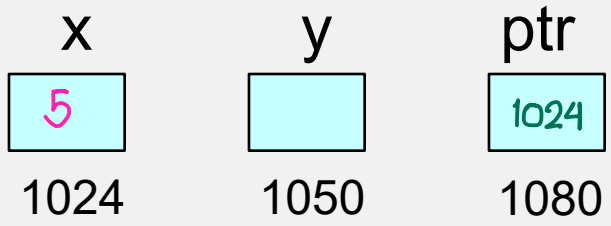
&y = Address = 1050

&ptr = Address = 1080

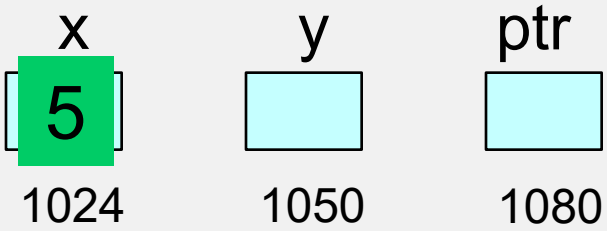
*ptr : 5

ตัวอย่าง 4.1

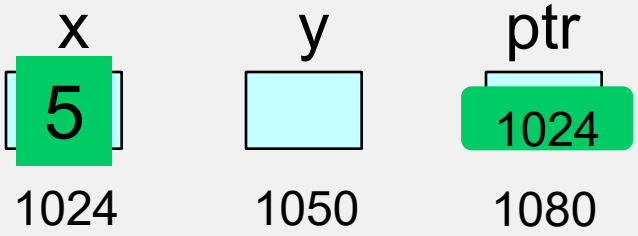
1. int x, y, *ptr;



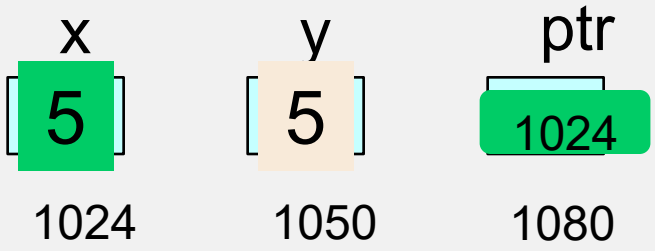
2. x=5;



3. ptr = &x;



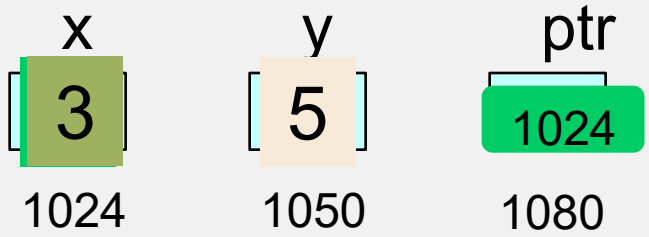
4. y = *ptr;



3 เข้าไปใส่ตำแหน่ง 1024

5. *ptr = 3;

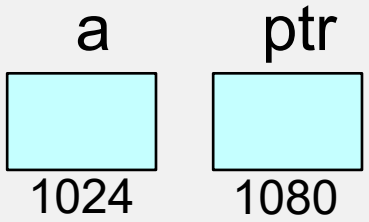
defined new value



Exercise 1

1. ประกาศตัวแปร ชื่อ a มี type float และประกาศตัวแปรชื่อ ptr มี type pointer to float กำหนดให้ a มีค่า 3.14

```
float a = 3.14 , *ptr ;
```



2. กำหนดให้ ptr เก็บตำแหน่งของ a

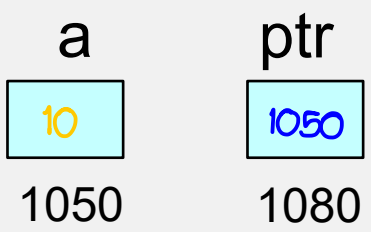
```
ptr = &a ;
```

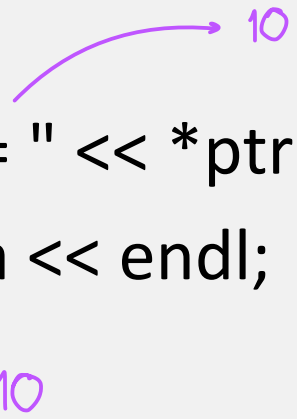
3. พิมพ์ค่า 3.14 โดยผ่านทาง ptr

```
cout << *ptr ;
```

ตัวอย่าง 4.2 ทดลองพิมพ์ค่า จากตัวแปร pointer

```
1. #include <iostream>
2. using namespace std;
3. int main()
4. {    int a=10 ,*ptr;
5.       ptr = &a;
6.       cout << "*ptr        = " << *ptr << endl;
7.       cout << "a    = " << a << endl;
8. }
```



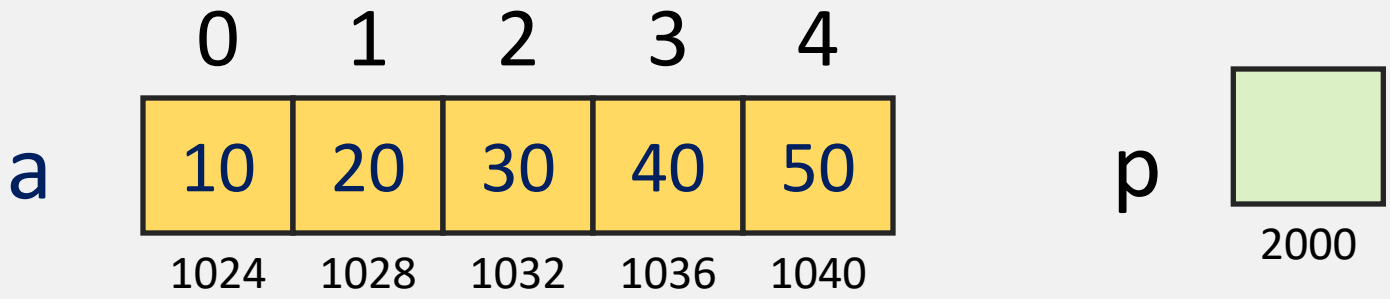


4.2 Pointer advanced

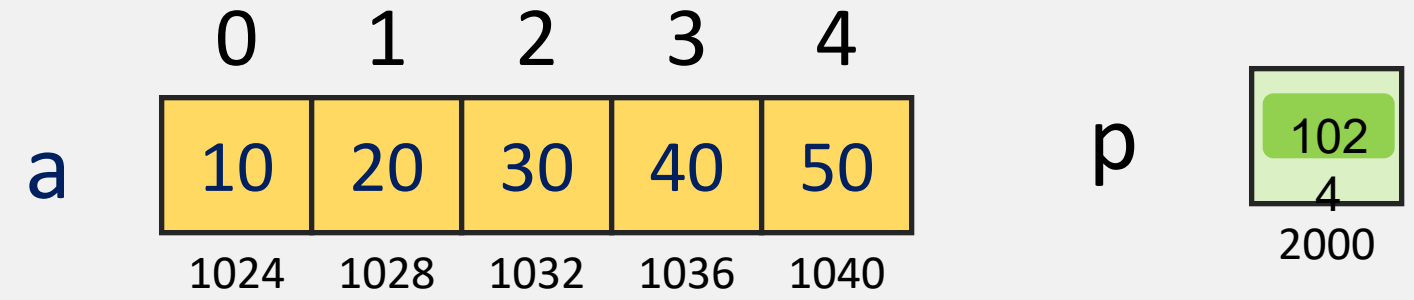
4.2.1 Pointer and array

คุณสมบัติของ Array

```
int  a[5]={10,20,30,40,50};
int  *p;
```



x : 5



a = 1024

a+1 = 1028

a[0] = 10

a[1] = 20

*a = 10

*(a+1)= 20

*a+1 = 11

&a[0] = 1024

a++

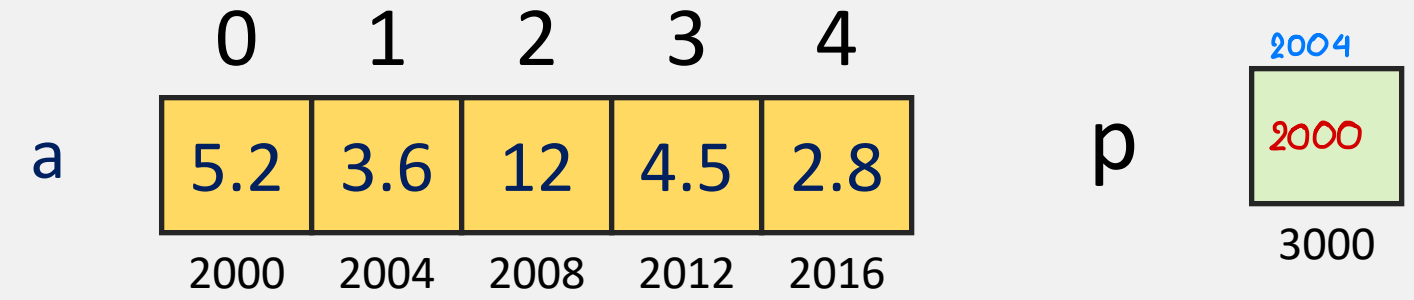
p = a (1024)

p++ = 1028

*p = 20

*(p++)= 30 (ต่อของเก่า)

ห้ามขยับ array



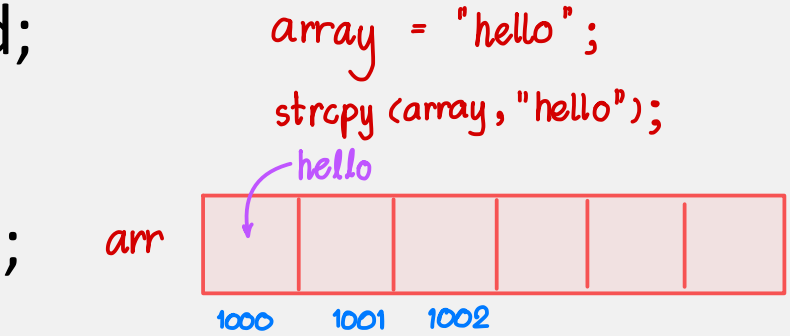

```
float a[5]={5.2,3.6,12,4.5,2.8};
float *p=a;
```

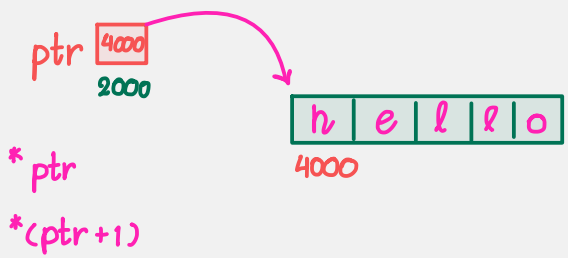
a	= 2000	*a	= 5.2	p	= a = &a
a+2	= 2008	*(a+2)=	12	p++	= 2004
a[0]	= 5.2	*a+1	= 6.2	*p	= 3.6
a[3]	= 4.5	*(a++)=	ไม่ได้ error	*(p++)=	12 (ต่อของเก่า)
		&a[3]	= 2012		

เป็น pointer

ตัวอย่าง 9.3 ความแตกต่างระหว่าง Array กับ Pointer to array

```
#include <iostream>
using namespace std;
int main()
{       char array[10];
        char *ptr;
        array="hello";   /*wrong */
        ptr = "hello";
        cout << ptr;
}
```

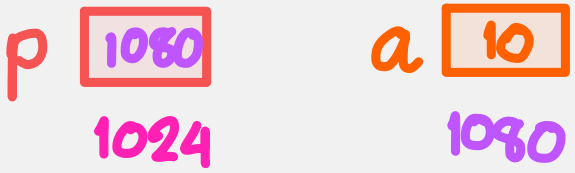




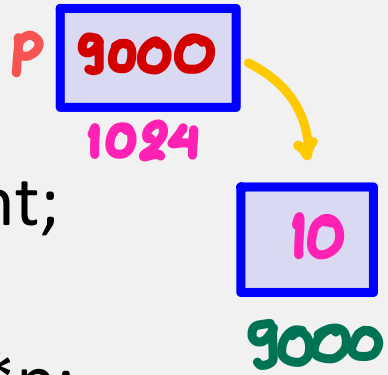
ตัวอย่าง 9.3

```
#include <iostream>
using namespace std;
int main()
{       int *p ,a;
        a=10;
        p=&a;
        cout << *p;
}
```

10

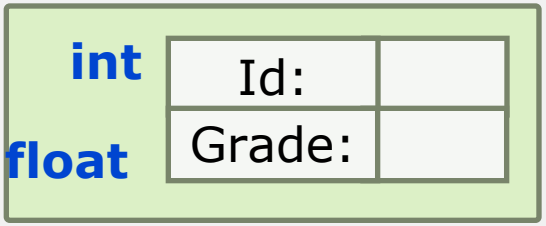


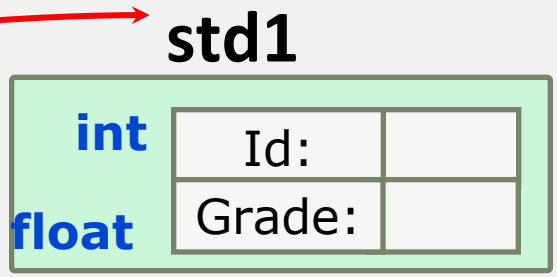

```
#include <iostream>
using namespace std;
int main()
{       int *p;
        p=new int;
        *p=10;
        cout << *p;
}
```

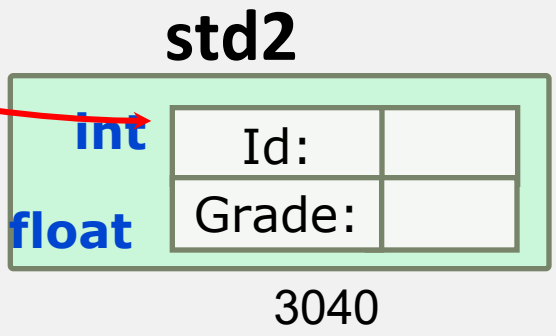


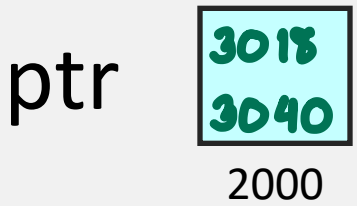
4.3.2 Pointer to Structure

```
struct record
{  4byte int id;
   4byte float grade;
};
struct record std1,std2;
struct record  *ptr;
```









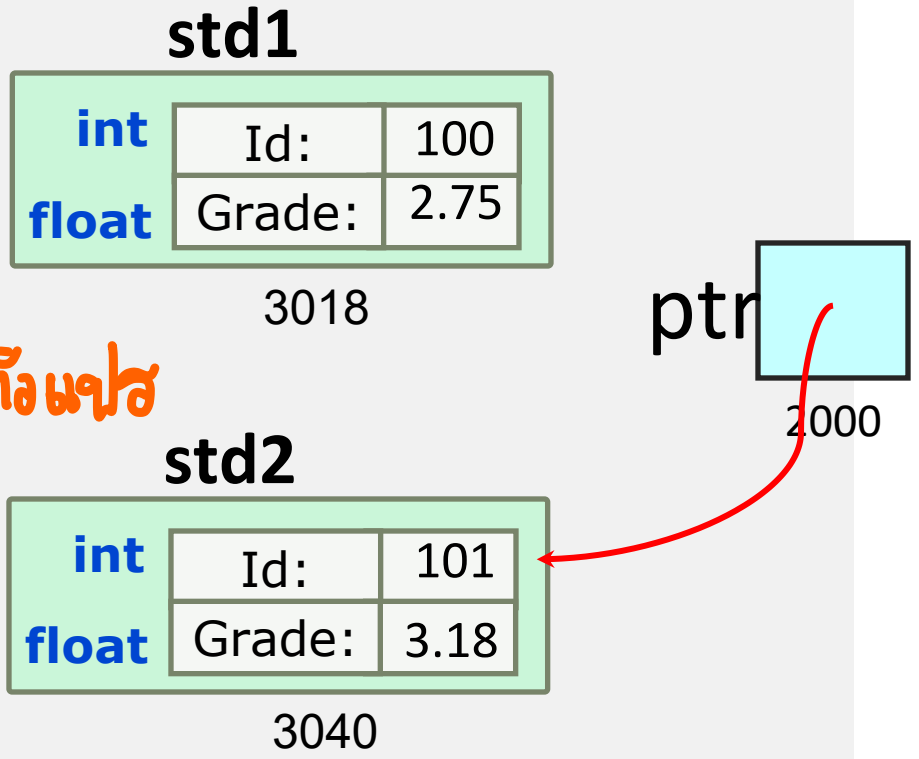
```
struct record std1, std2;
struct record  *ptr;

std1.Id=100;
std1.Grade=2.75;

ptr=&std2;
(*ptr).Id=101;   หรือ
ptr->Id=101;
ptr ->Grade=3.18;
```

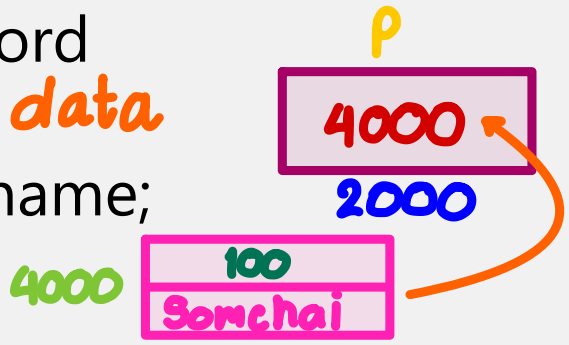
ใส่ค่าผ่านตัวแปร

ใส่ค่าผ่าน pointer

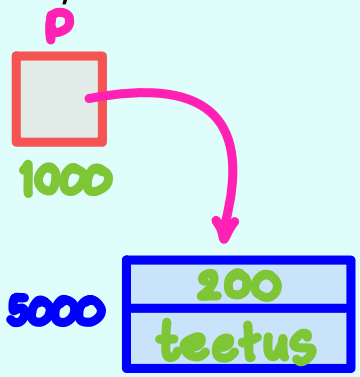



```
ptr → Id = 101;
ptr → Grade = 3.18;
```

```
#include <iostream>
using namespace std;
struct record
{   int id;
    string name;
};
int main()
{   struct record  data, *p;
    data.id=100;
    data.name="Somchai";
    p=&data;
    cout << p->id <<
     " " << p->name << endl;
}
```




```
#include <iostream>
using namespace std;
struct record
{   int id;
   string name;
};
int main()
{  struct record  *p;
   p = new struct record;
   p->id =200;
   p->name="Teetus";
   cout << p->id << " "
   << p->name << endl;
}
```



```
#include <iostream>
using namespace std;
struct record
{    int value;
     struct record *next;
};
```

```
int main()
{         struct record *n1;
          n1=new struct record;
          n1->value=10;
          n1->next=NULL;
          cout << n1->value <<endl;
}
```