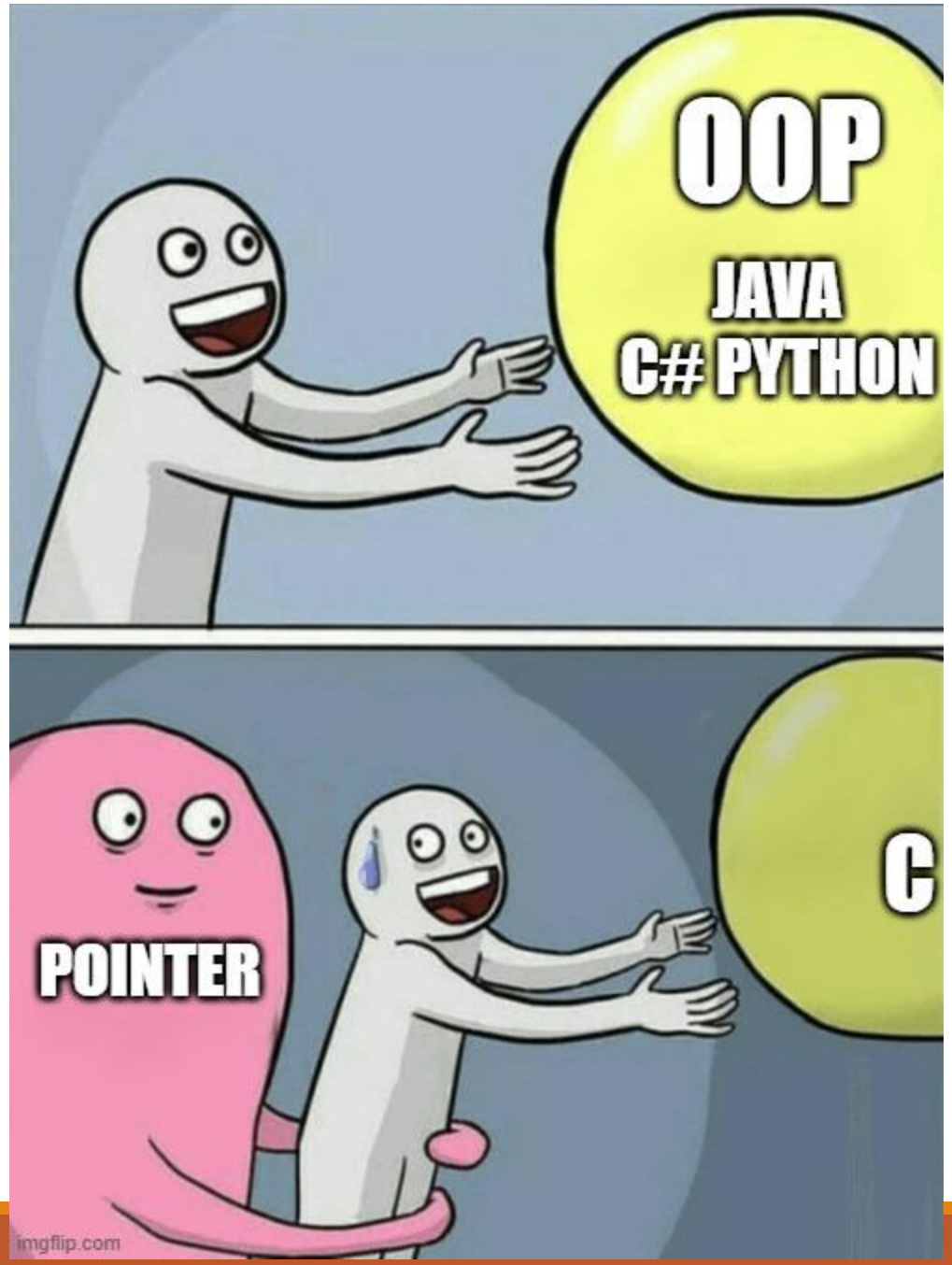
OOP & data struct

7. Pointer

BY SOMSIN THONGKRAIRAT


OOP
JAVA
C# PYTHON
POINTER
C
imgflip.com

Pointer

- A variable that use to point to another
- ตัวแปรที่ ชี้ ไปยังตัวแปรอื่น

Boring?

- learn this for a thousand time, boring now?
- เรียน pointer มานับครั้งแล้วไม่ถ้วย เบื่อกันหรือยังครับ ^^

What is pointer / อะไรคือ pointer

- a variable that store <u>address</u> of variable

- ตัวแปรที่ใช้เก็บ <u>ที่อยู่</u> ของตัวแปร

What is gasoline 95 / อะไรคือ แก๊สโซฮอล์ 95

น้ำมัน เบนซิน 95 ผสมกับเอทิลแอลกอฮอล์ บริสุทธิ์ 99.5% (อัตราส่วน เบนซิน 90% : เอทานอล 10%) เพื่อทดแทนสาร MTBE (Methyl Tertiary Butyl Ether) เผาไหม้ได้ดี ใช้ทดแทนเบนซิน 95 ได้ และมีราคาถูกกว่า

Gasohol 95 is a type of fuel in the group of gasoline. It contains 9 parts of 95 octane gasoline and one part of ethyl alcohol that provides driving performance or response as fast as the 95 gasoline itself, suitable for all models of Stallions.

Who care?

ใครสน?

OOP & data struct

7. Pointer with class

BY SOMSIN THONGKRAIRAT

ตัวแทน / representative

Java -> instance

C# - > reference types

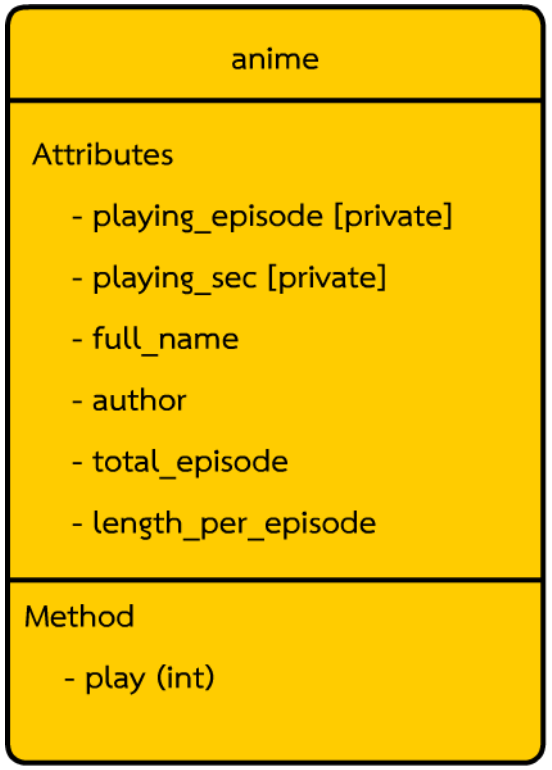
C++ -> Pointer

Representative? / ตัวแทน?

อะไรคือตัวแปรแบบธรรมดา ? / what normal variable do ?

- it stores the value itself

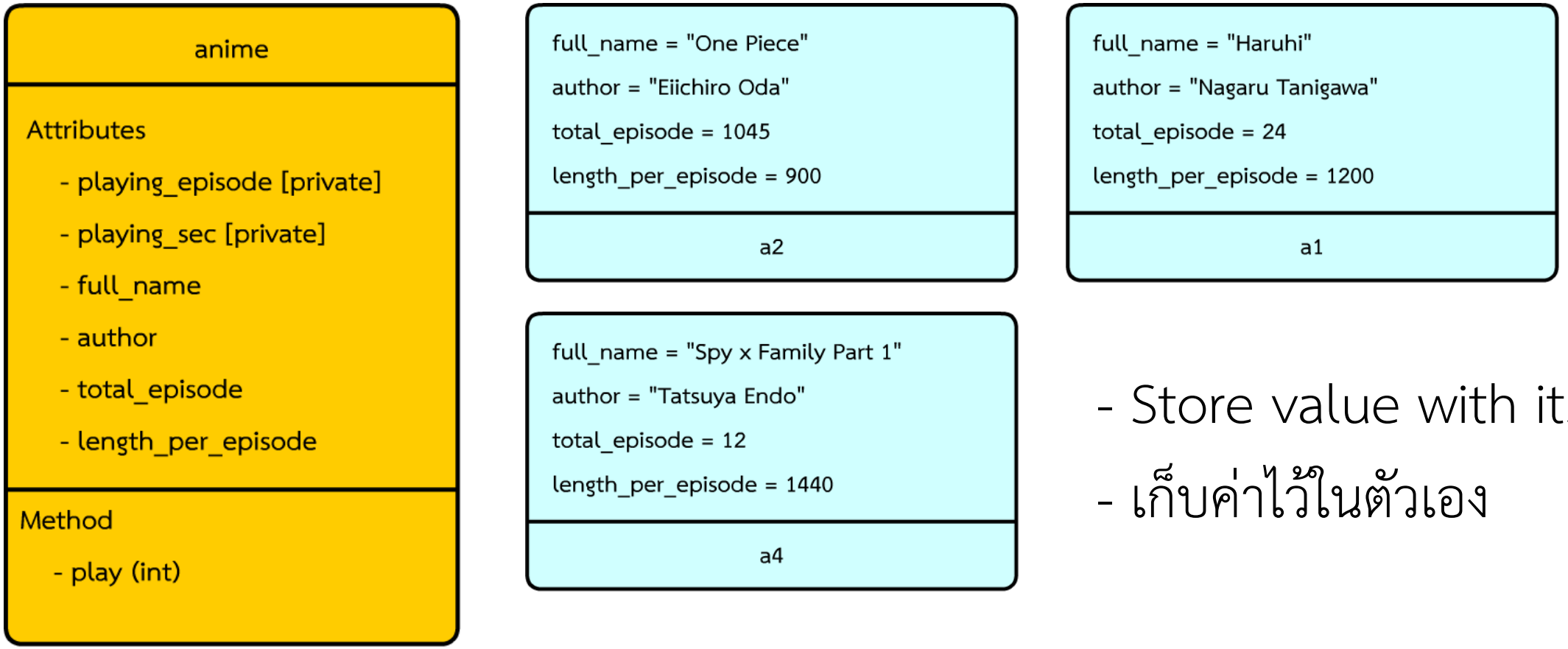
- เก็บข้อมูลไว้ในตัวมันเอง



Example

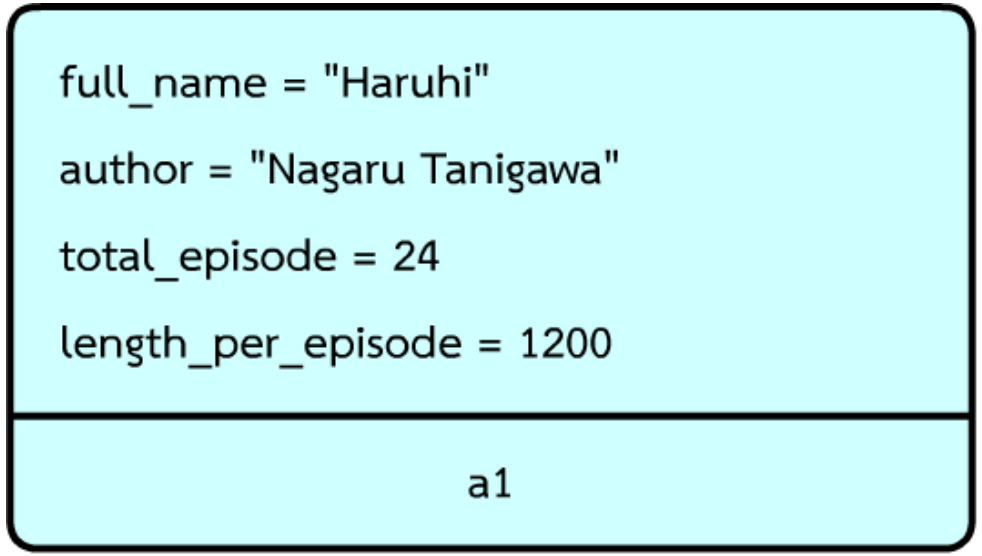
```
anime a1,a2;
a1.full_name = "The Melancholy of Haruhi Suzumiya";
a1.author = "Nagaru Tanigawa";
a1.total_episode = 24;
a1.length_per_episode = 1200;
```

Example of variable or object



- Store value with itself
- เก็บค่าไว้ในตัวเอง

Representative benefit / ประโยชน์ตัวแทน



a1 ≈ 23 byte

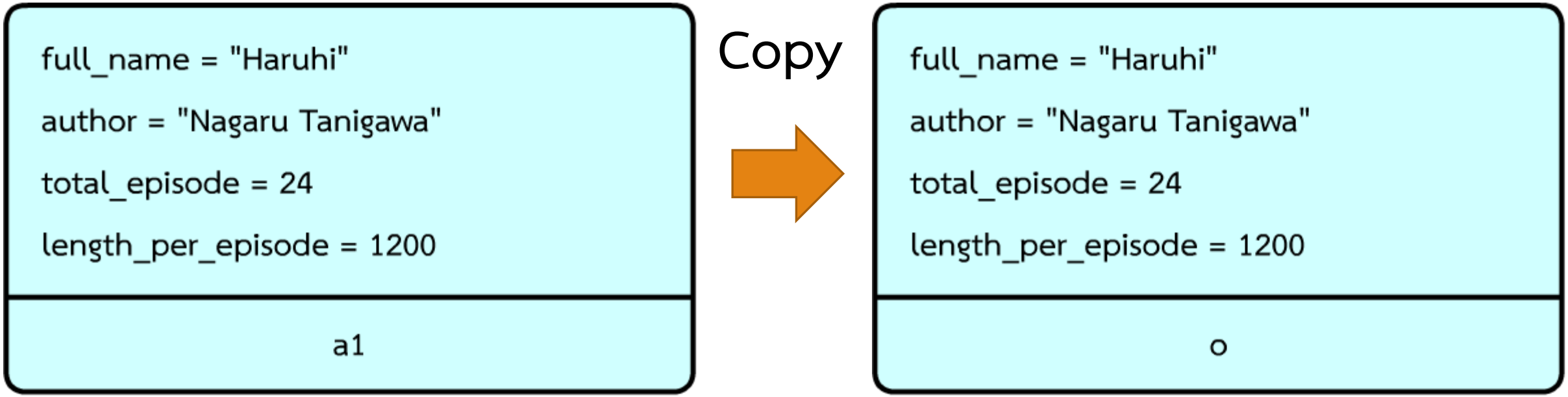
If not Representative / ถ้าไม่มีตัวแทน

```
anime *p;
anime o;
p = &a1;
o = a1;
cout << p->full_name << endl;
cout << calculate_text_size(o.full_name) << endl;
cout << calculate_text_size(p->full_name) << endl;
```

p is representative of a1

P คือตัวแทนของ a1

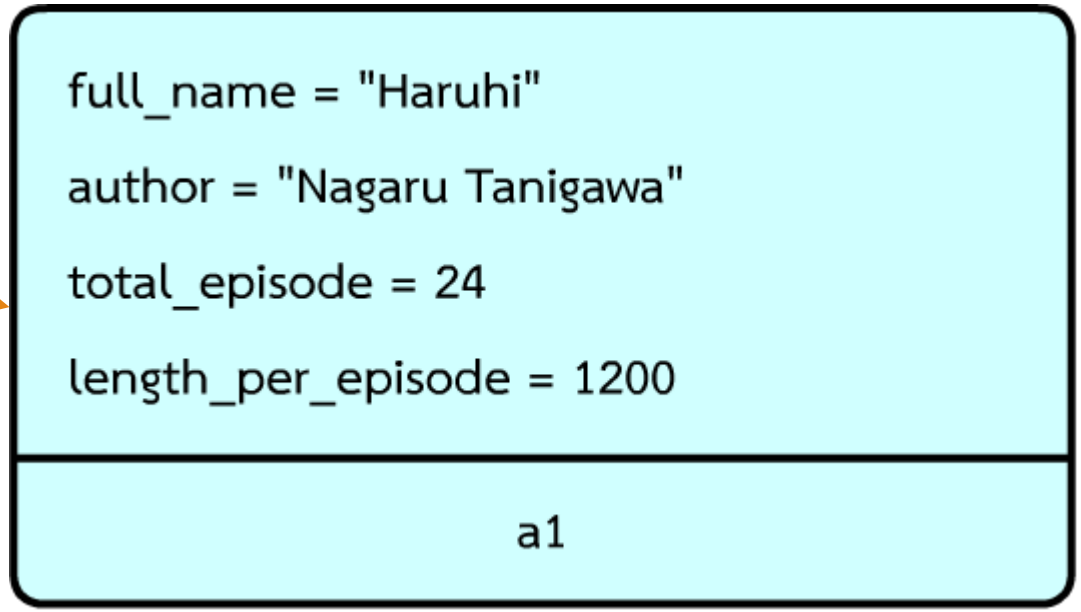
```
o = a1;
```



And process

```
p = &a1;
```

P



process!

Same result!

What difference / ต่างกันอย่างไร

- Same result!
- memory?
- convenience?

Syntax pointer declaration

(*) Follow by name / เครื่องหมาย * ตามด้วยชื่อ

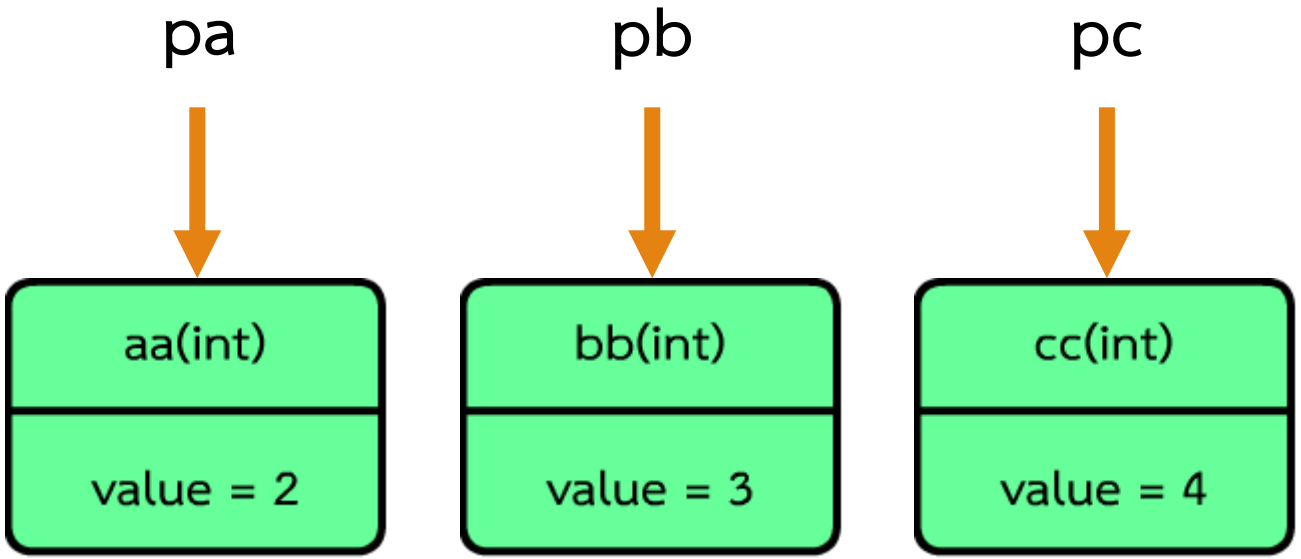
```
anime *p,q,r,*s;
anime *t,u;
```

Pointer : p,s,t

Object : q,r,u

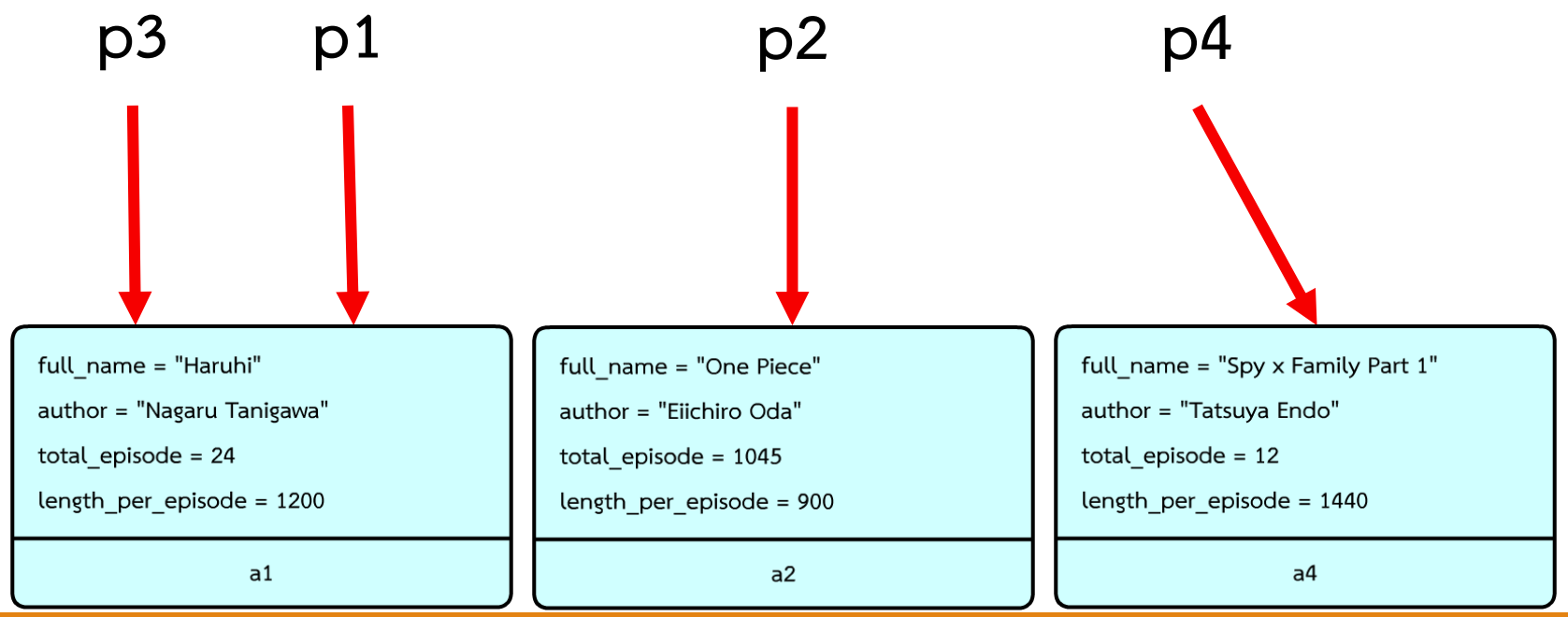
Syntax pointer assign (variable)

```
int *pa,*pb,*pc;
int aa = 2,bb = 3,cc =4;
pa = &aa;
pb = &bb;
pc = &cc;
```



Syntax pointer assign (object)

```
anime *p1,*p2,*p3,*p4;
p1 = &a1;
p3 = &a1;
p2 = &a2;
p4 = &a4;
```



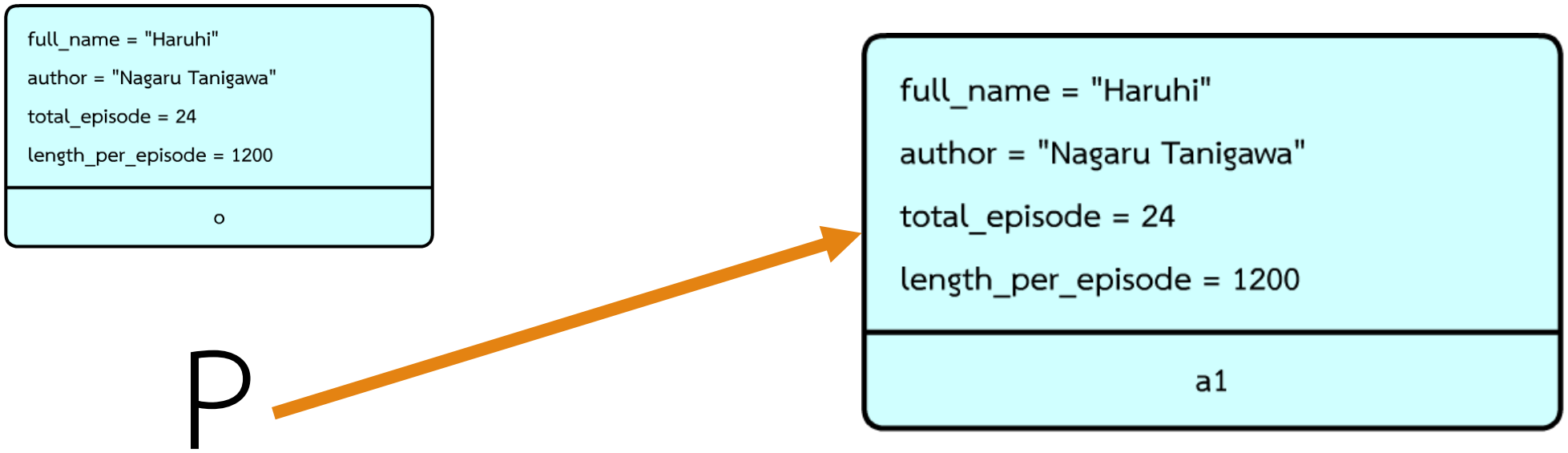
Syntax pointer usage

Normal object using (.) operator

representator object using (->) operator

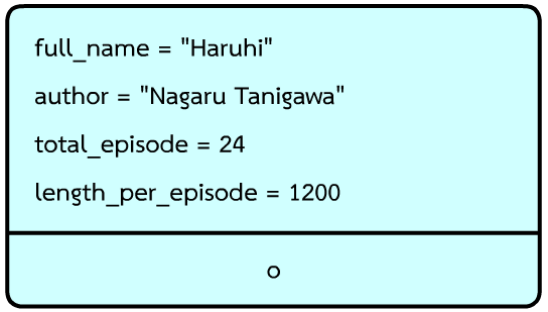
```
p = &a1;
o = a1;
cout << a1.full_name << endl;
cout << calculate_text_size(o.full_name) << endl;
cout << calculate_text_size(p->full_name) << endl;
```

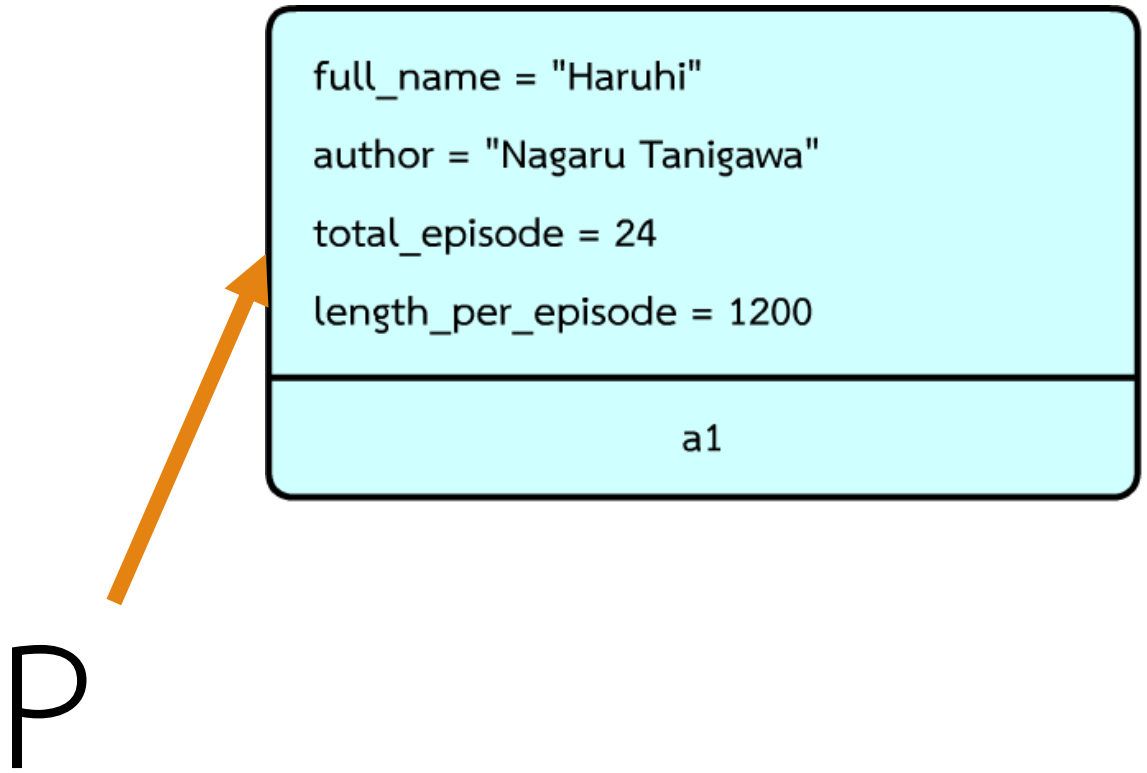
```
anime a1("Haruhi","Nagaru Tanigawa",24,1200);
p = &a1;
o = a1;
cout << a1.full_name << endl;
cout << calculate_text_size(o.full_name) << endl;
cout << calculate_text_size(p->full_name) << endl;
```



Benefit / ข้อดี

- no extra memory
- pass by reference



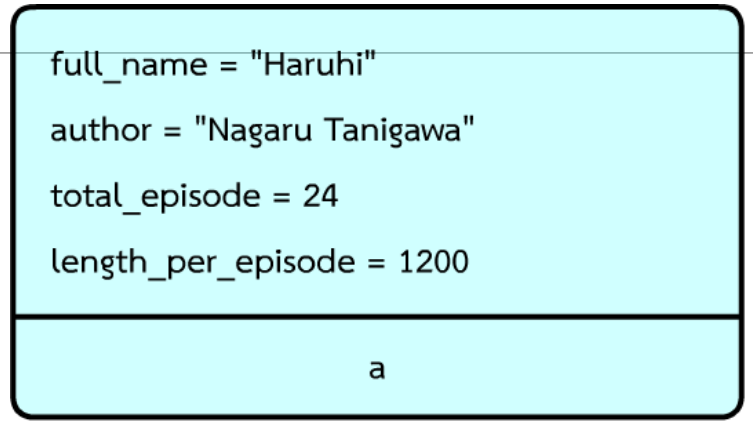


Quiz

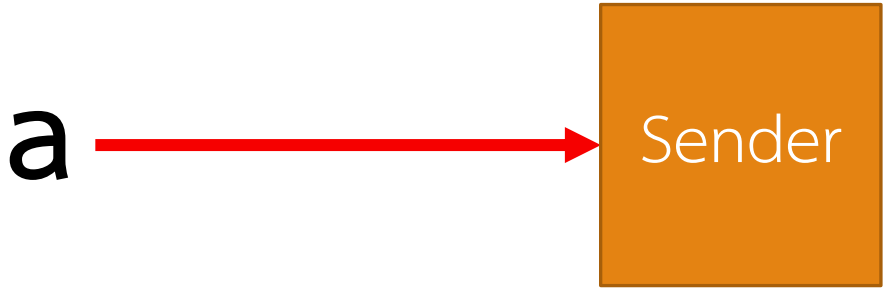
CHANGE OUTPUT

Pass by reference

```
void add_episode(anime a){
    a.total_episode++;
}
```




```
void add_episode(anime *a){
    a->total_episode++;
}
```



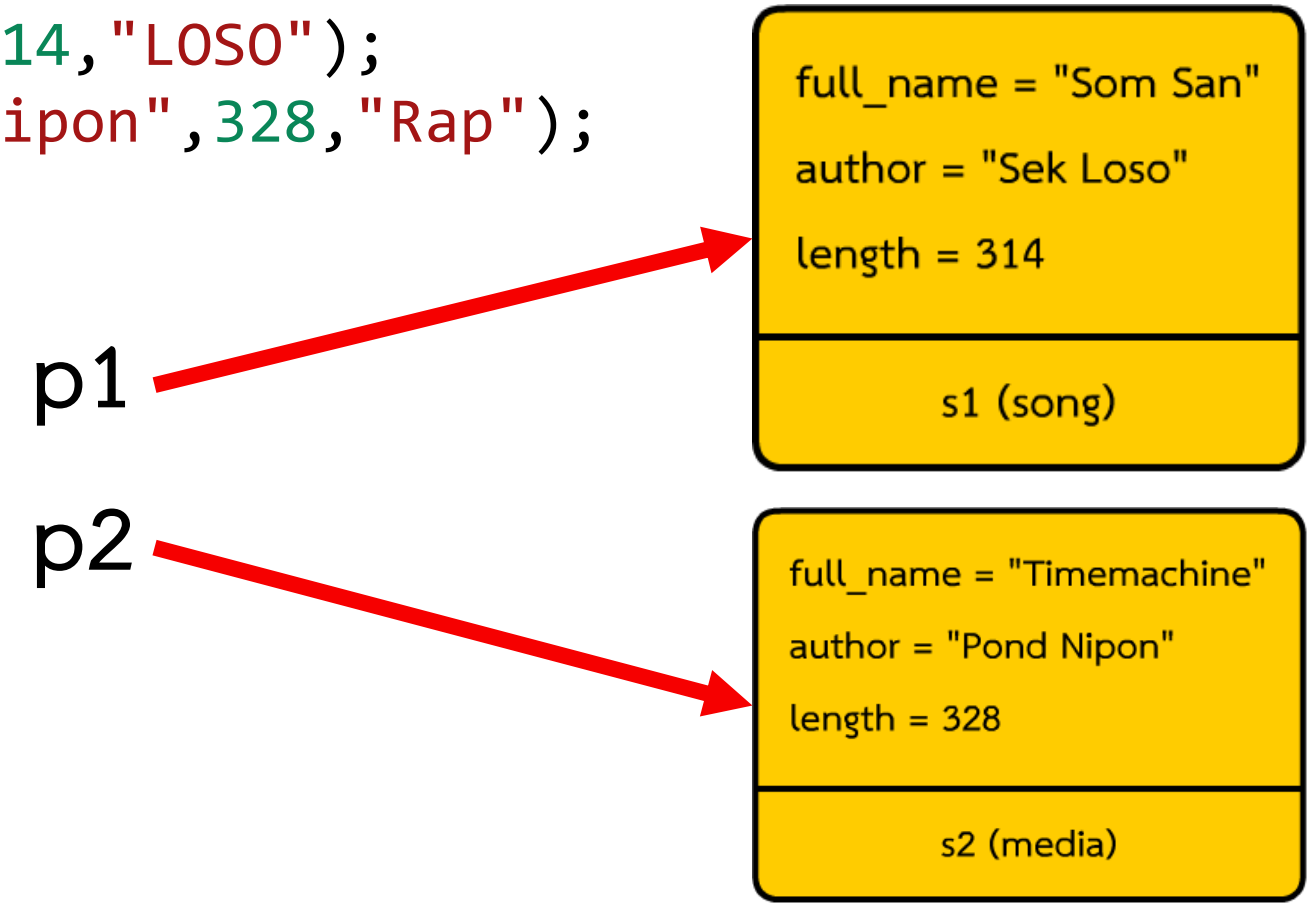
```
add_episode(a1);
add_episode(o);
add_episode(p);

cout << a1.total_episode << " ";
cout << o.total_episode << " ";
cout << p->total_episode << endl;
```

```
song s1("Som San","sek loso",314,"LOSO");
song s2("Timemachine ","Pond Nipon",328,"Rap");

song *p1,*p2;
p1 = &s1;
p2 = &s2;

s1.play(1);
s2.play(2);
p1->play(4);
p2->play(8);
```




```
cout << s1.get_playing_sec() << "," << s2.get_playing_sec() << endl;
```

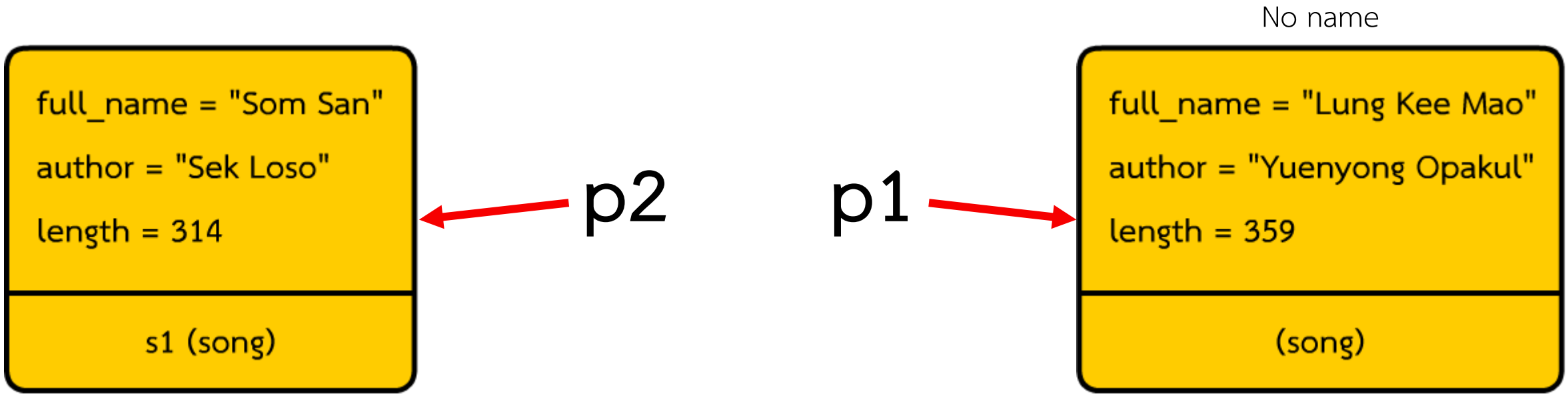
New object

```
pointer = new datatype(constructor);
```

```
p1 = new song("Lung Kee Mao","Yuenyong Opakul",359,"Carabao");
```

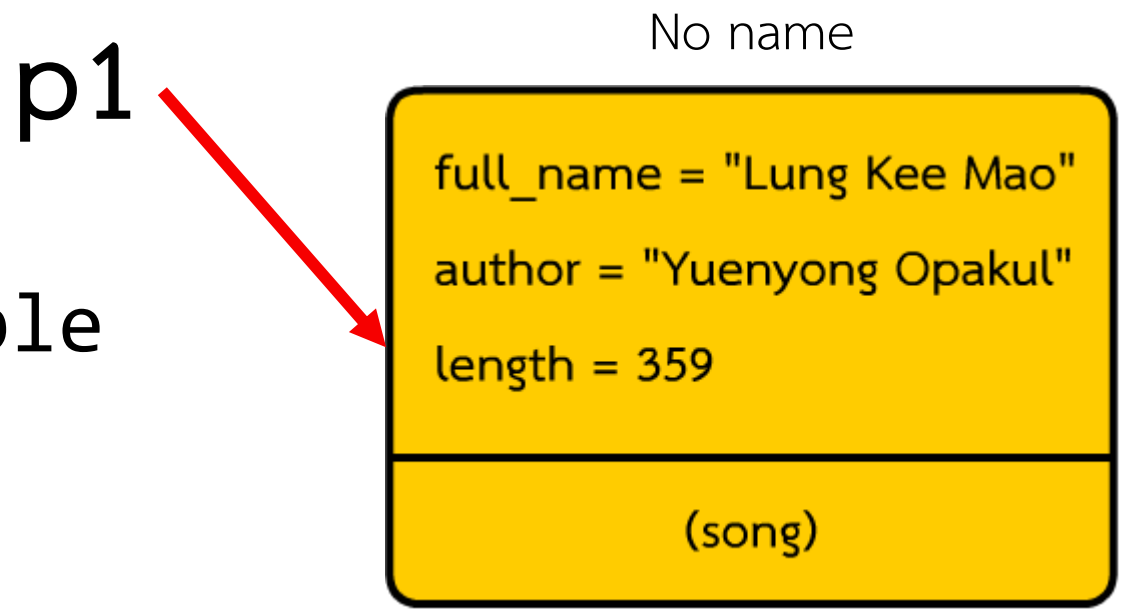
- build new object in memory
- สร้าง object ใหม่ขึ้นมา

```
song s1("Som San","sek loso",314,"LOSO");
song *p1,*p2;
p2 = &s1;
p1 = new song("Lung Kee Mao","Yuenyong Opakul",359,"Carabao");
```



```
p1 = new song("Lung Kee Mao","Yuenyong Opakul",359,"Carabao");

p1->play(16);
p1->print_song();
// now we playing Lung Kee Mao of Carabao at 16 sec
```



Can't access via variable

ไม่สามารถเข้าถึงได้โดยตัวแปร

Delete object

```
delete pointer;
```

- destroy object free memory to system
- ทำลาย object และคืน memory

```
p1 = new song("Lung Kee Mao","Yuenyong Opakul",359);

p1->play(16);
p1->print_song();

delete p1;
```

p1 → NULL

p1 → NULL

```
delete p1;
p1->play(32); // crash!
p1->print_song(); // crash!
```

To detect null pointer

- assign null to pointer after delete
- กำหนดค่าให้เป็น null หลัง delete

```
delete p1;
p1 = NULL;
```

```
delete p1;
p1 = NULL;
//p1->play(32); // crash!
//p1->print_song(); // crash!

if(p1 == NULL){
    cout << "no object" << endl;
}
else{
    cout << "object name : " << p1->get_name() << endl;
}
```

Conclude

- representation
- new , delete keyword

LAB

สร้าง class playlist ขึ้นมาเพื่อเก็บ song โดยมี

- method add(*song) เพื่อ เพิ่ม song เขาไปใน playlist
- method add(string,string,int,string) เพื่อเพิ่ม song ใหม่ตาม constructor
- method remove() เพื่อ ลบเพลงสุดท้ายที่อยู่ใน list
- method play_all() เพื่อ แสดงเพลงทุกเพลงที่อยู่ใน list
- method get_song(int n) เพื่อ return pointer ของ song ที่อยู่ใน list ลำดับที่ n หากไม่มีให้ return NULL

* Song จะมีไม่เกิน 4 song ใน เดียวกัน


Start
S1
S3
S4
ADD S5
S1
S3
S4
S5*
remove
S1
S3
S4
remove
S1
S3
remove
S1
ADD s1
S1
S1