

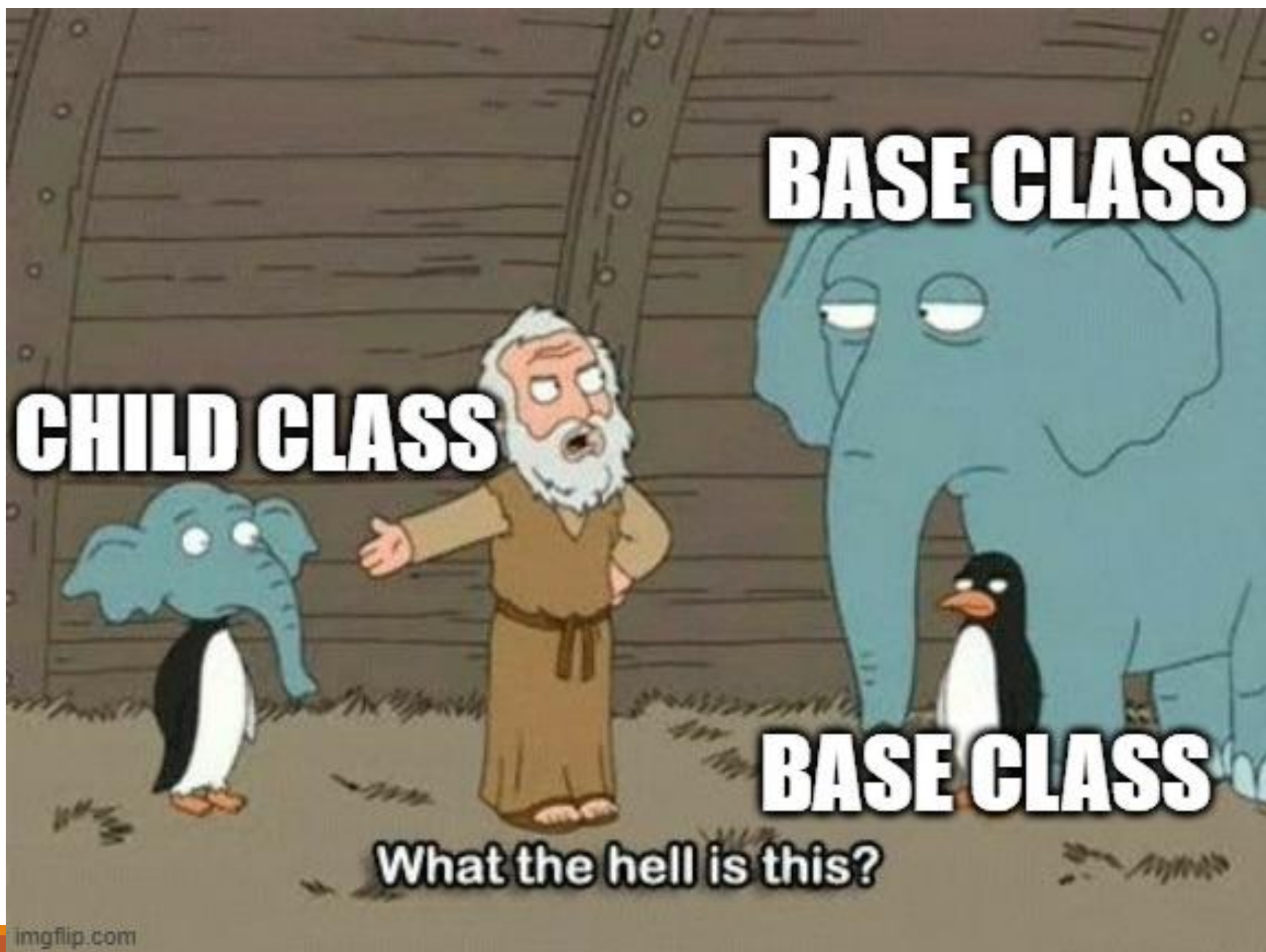
# OOP & data struct

## 3. Inheritance

---

BY SOMSIN THONGKRAIRAT





## Class Con.

---

### Media class (1 media)

#### Properties

- Public -> name, length // ชื่อ, ความยาว (sec)
- Private -> playing\_sec // เวลาล่าสุดที่เล่น (pause)

#### Method – play(int)

```

4
5 ~ class media{
6     private :
7         int playing_sec; // last played seconds in episode
8
9     protected :
10         string name;
11         int length; // media length in Seconds
12
13     public :
14 ~     void play(int time){ // play method
15         int remaining_time = length - playing_sec;
16         cout << "Playing " << name << " at[" << playing_sec <<"] : [" << remaining_time << " sec] remaining"
17 ~         if(time > remaining_time){
18             playing_sec = 0;
19         }
20 ~         else{
21             playing_sec += time;
22         }
23     }
24

```

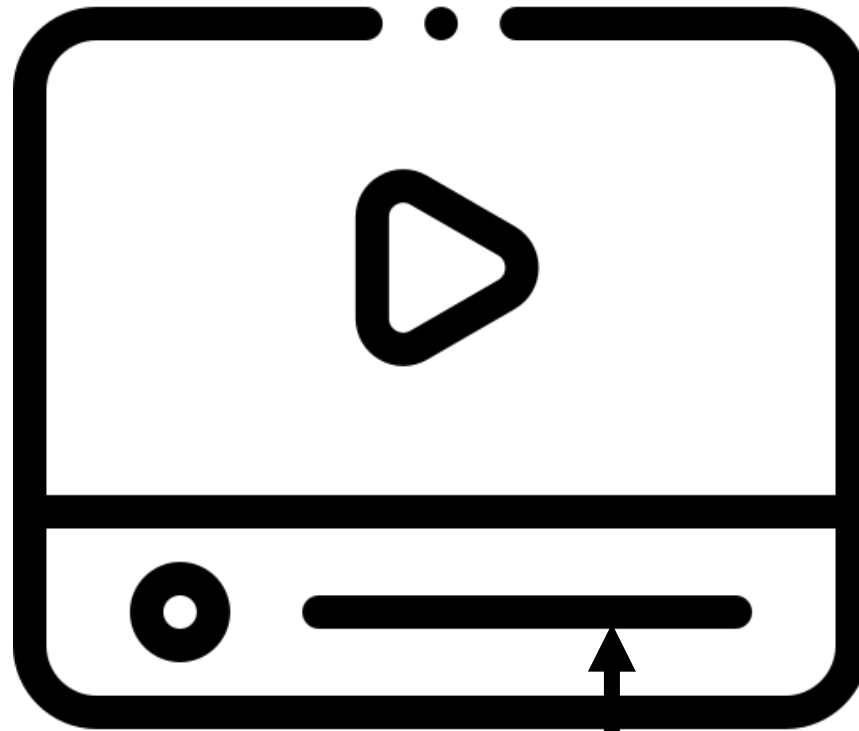
# Media class Constructor

---

```
25     media(){ // default constructor
26         name = "unknow";
27         length = 0;
28         playing_sec = 0;
29     }
30
31     media(string _name,string _author,int _length){ // 3 parameter constructor
32         name = _name;
33         length = _length;
34         playing_sec = 0;
35     }
36
```

# Media Class

---



Playing\_sec

## media

### Attributes

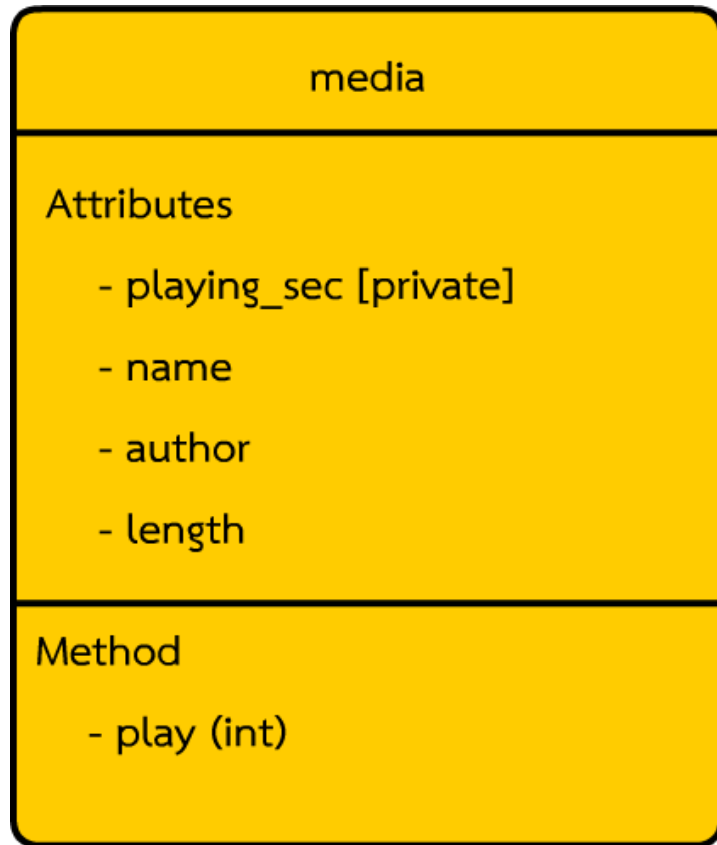
- playing\_sec [private]
- name
- author
- length

### Method

- play (int)

# Everything look good!

---



Can handle

- Movie
- Song
- Episode



# One class handle every classes

---

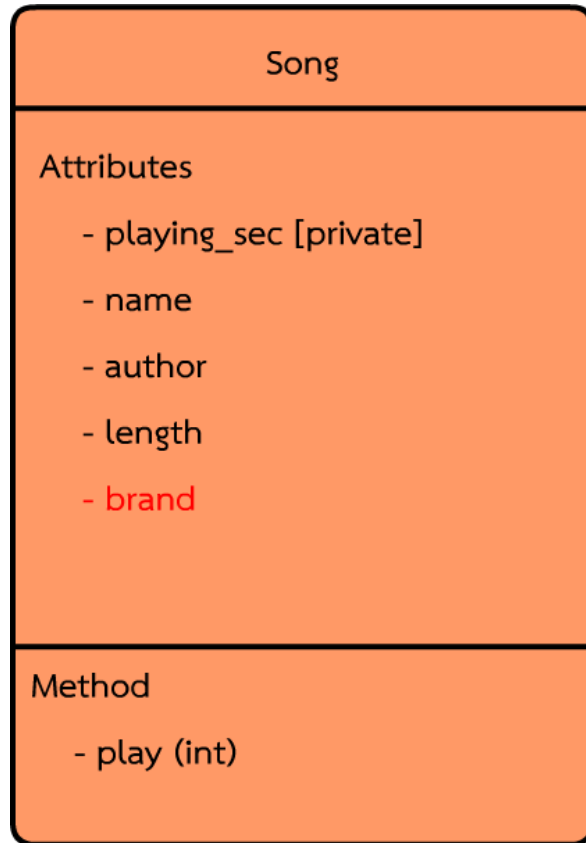
```
int main(){

    media s1("Som San","sek loso",314);
    media s2("Timemachine ","Pond Nipon",328);
    media m1("The Disappearance of Haruhi Suzumiya","Nagaru Tanigawa",9707);
    media m2("Avatar","James Cameron",9720);
    media e1("Thi da Satan : ep 3 Earth Water Wind Fire","Kantana",3600);
    media e2("Start-Up (2020) : ep 16 Scale up","Studio Dragon",4800);

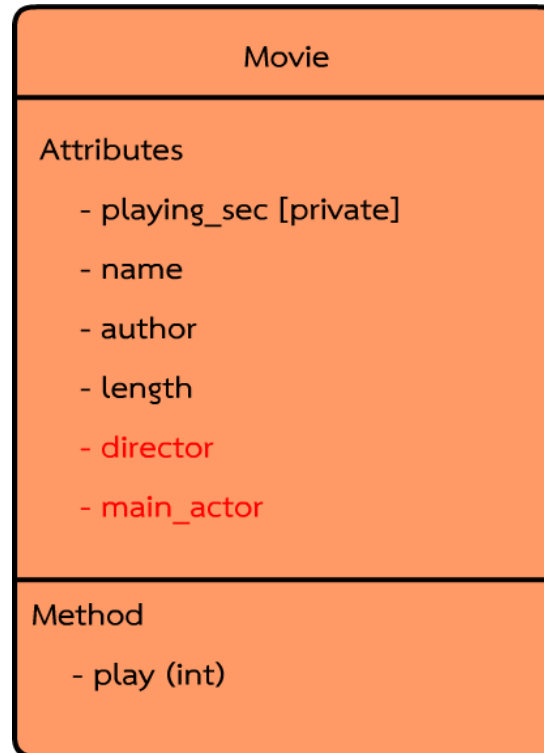
    s1.play(200);
    s1.play(200);
    s1.play(200);

    return 0;
}
```

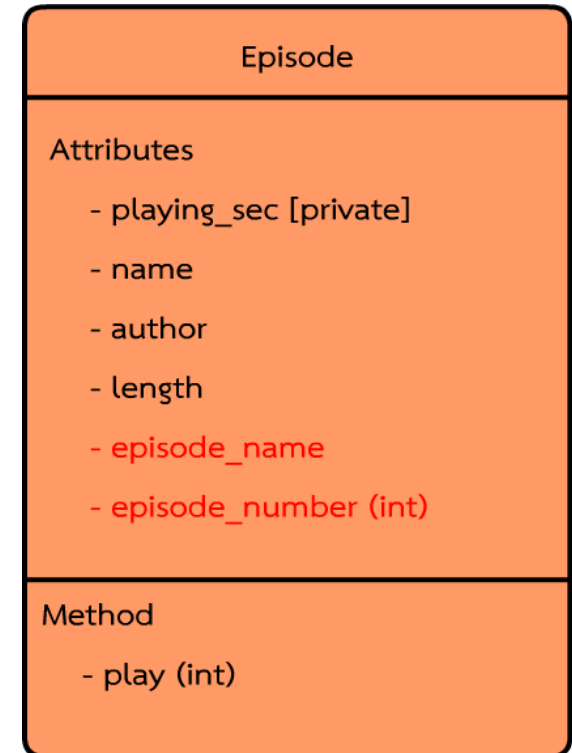
# Make it more specific



Add Brand



Add director , actor



Add episode name and num

# Hard code method (create 3 Classes)

```
41 class song{
42 private :
43     int playing_sec; // last played seconds in episode
44
45 public :
46     string name;
47     string author;
48     string brand; // added from media
49     int length; // media length in Seconds
50
51     void play(int time){ // play method
52         int remaining_time = length - playing_sec;
53         cout << "Playing " << name << " at[" << playing_sec <<
54         if(time > remaining_time){
55             playing_sec = 0;
56         }
57         else{
58             playing_sec += time;
59         }
60     }
61
62     song(){ // default constructor
63         name = "unknown";
64         length = 0;
65         playing_sec = 0;
66     }
67
68     song(string _name,string _author,int _length){ // 3 param
69         name = _name;
70         author = _author;
```

```
77 class movie{
78 private :
79     int playing_sec; // last played seconds in episode
80
81 public :
82     string name;
83     string author;
84     string director;
85     string main_actor;
86     int length; // media length in Seconds
87
88     void play(int time){ // play method
89         int remaining_time = length - playing_sec;
90         cout << "Playing " << name << " at[" << playing_sec <<
91         if(time > remaining_time){
92             playing_sec = 0;
93         }
94         else{
95             playing_sec += time;
96         }
97     }
98
99     movie(){ // default constructor
100         name = "unknown";
101         length = 0;
102         playing_sec = 0;
103     }
104
105     movie(string _name,string _author,int _length){ // 3 param
```

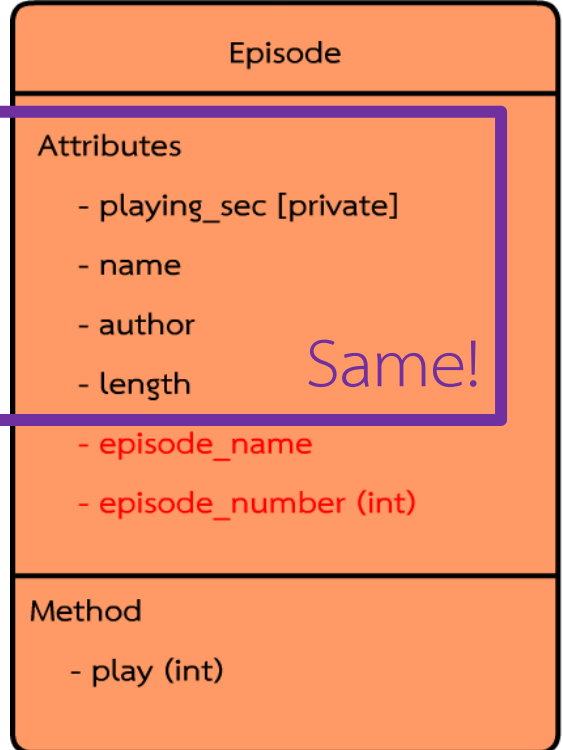
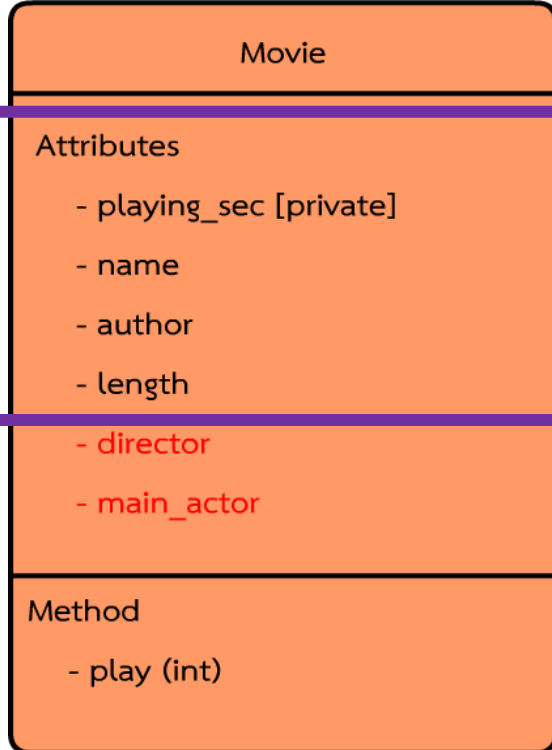
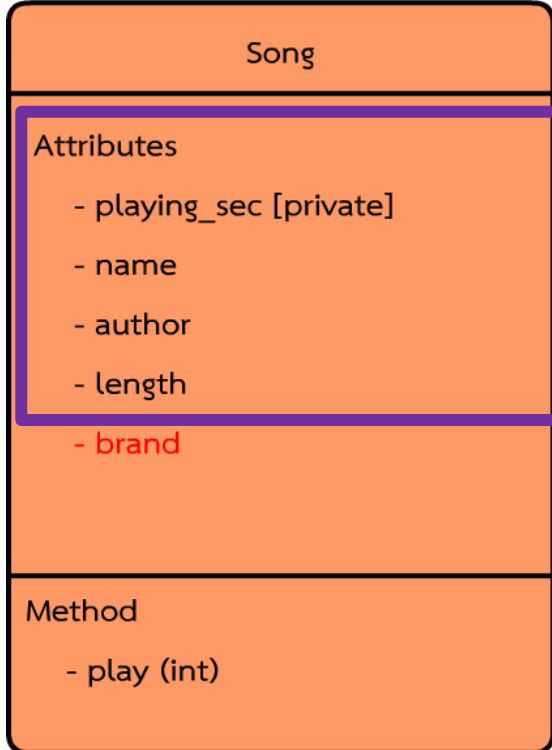
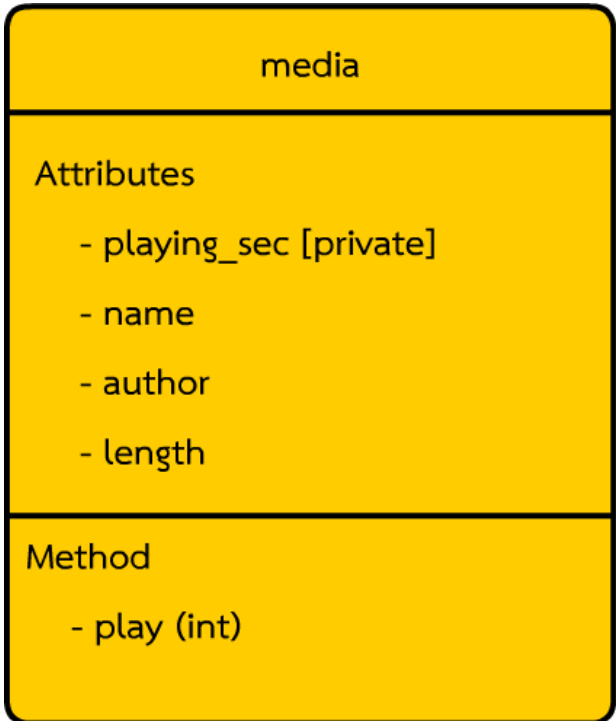
```
114 class episode {
115 private :
116     int playing_sec; // last played seconds in episode
117
118 public :
119     string name;
120     string author;
121     int episode_number;
122     string episode_name;
123     int length; // media length in Seconds
124
125     void play(int time){ // play method
126         int remaining_time = length - playing_sec;
127         cout << "Playing " << name << " at[" << playing_sec <<
128         if(time > remaining_time){
129             playing_sec = 0;
130         }
131         else{
132             playing_sec += time;
133         }
134     }
135
136     episode(){ // default constructor
137         name = "unknown";
138         length = 0;
139         playing_sec = 0;
140     }
141
142     episode(string _name,string _author,int _length,int _episode_number,
```

At least it work!

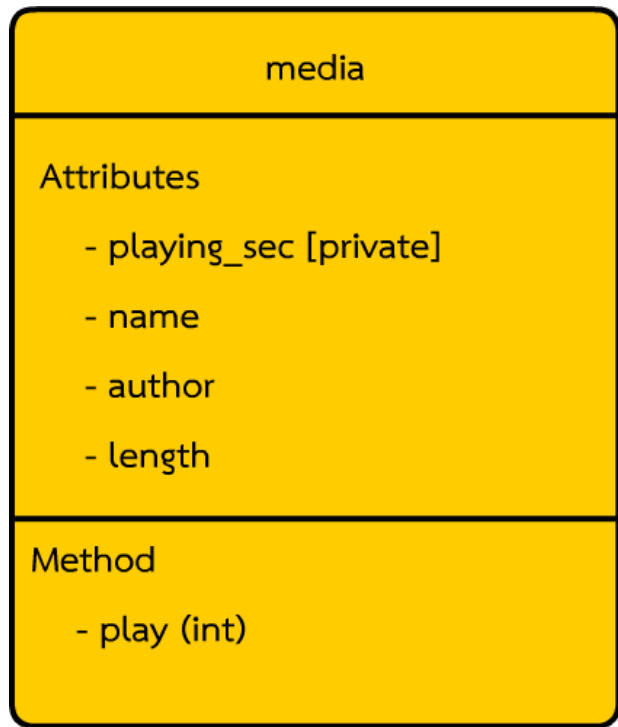
## Inheritance method

---

- Transfer class component to child class
- ส่งต่อส่วนประกอบไปยัง class อื่น (คลาสลูก)

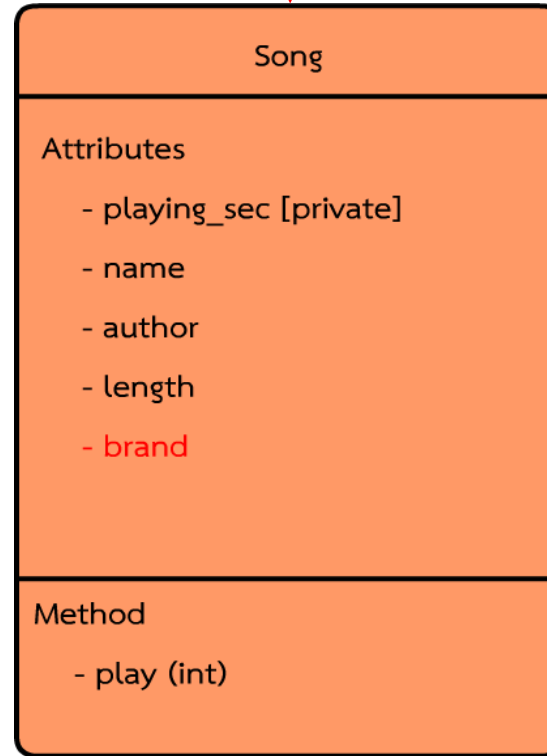


Same!

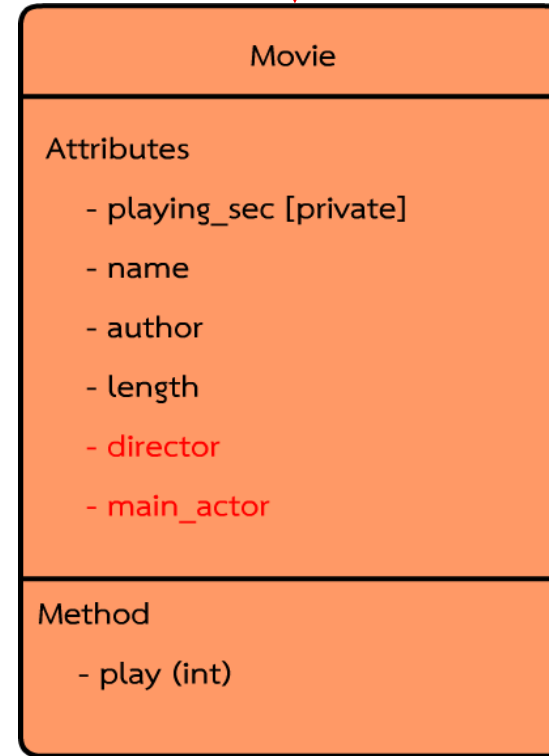


Parent Class

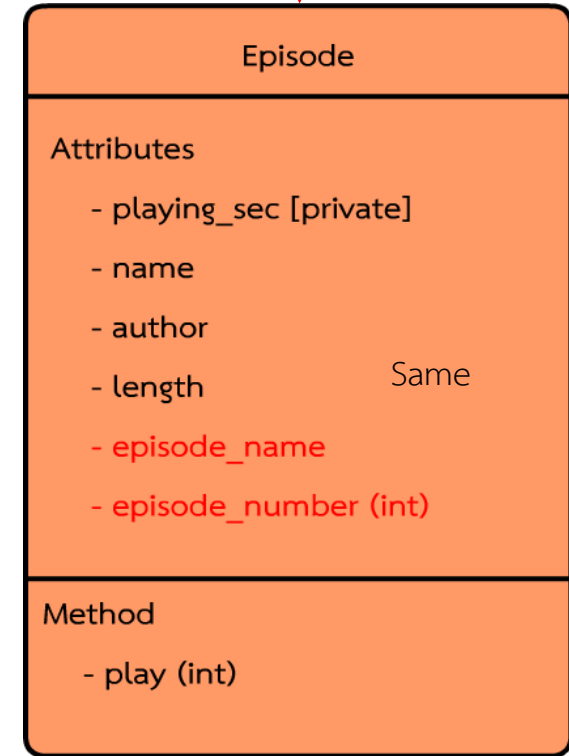
inherit (สืบทอด)



Child Class



Child Class



Child Class

# C++ Syntax

---

```
class [child_class_name] : [modifier1] [base_class_name1] , [modifier2] [base_class_name2] ...{  
    // class component  
}
```

```
class movie : public media{  
public :  
    string director;  
    string main_actor;  
};
```

```
class episode : private media{  
public :  
    int episode_number;  
    string episode_name;  
};
```

```
41 class song : public media{
42 public :
43     string brand;
44 };
45
46 class movie : public media{
47 public :
48     string director;
49     string main_actor;
50 };
51
52 class episode : public media{
53 public :
54     int episode_number;
55     string episode_name;
56 };
```

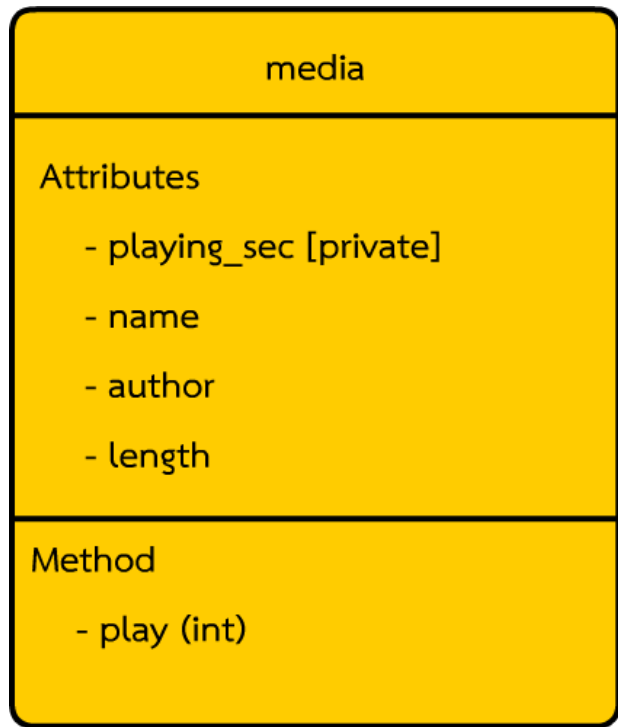




```
67 song s1;
68 s1.fullname = "Som San";
69 s1.author = "sek loso";
70 s1.length = 314;
71 s1.brand = "LOSO";
72 s1.play(10);
73
74 movie m1;
75 m1.fullname = "Avatar";
76 m1.author = "James Cameron";
77 m1.length = 9720;
78 m1.director = "James Cameron";
79 m1.main_actor = "Sam Worthington";
80 m1.play(20);
81
82 episode e1;
83 e1.fullname = "Thi da Satan : ep 3 Earth Water Wind Fire";
84 e1.author = "Kantana";
85 e1.length = 4800;
86 e1.episode_number = 3;
87 e1.episode_name = "Earth Water Wind Fire";
88 e1.play(30);
```

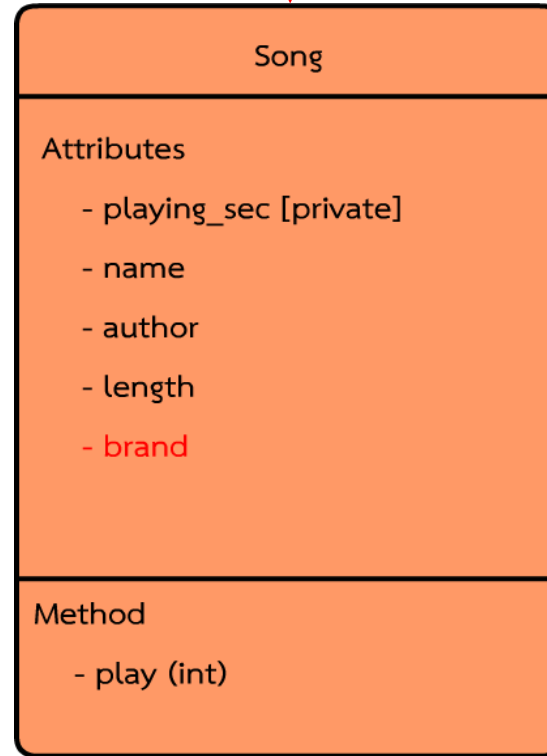


All can use play();

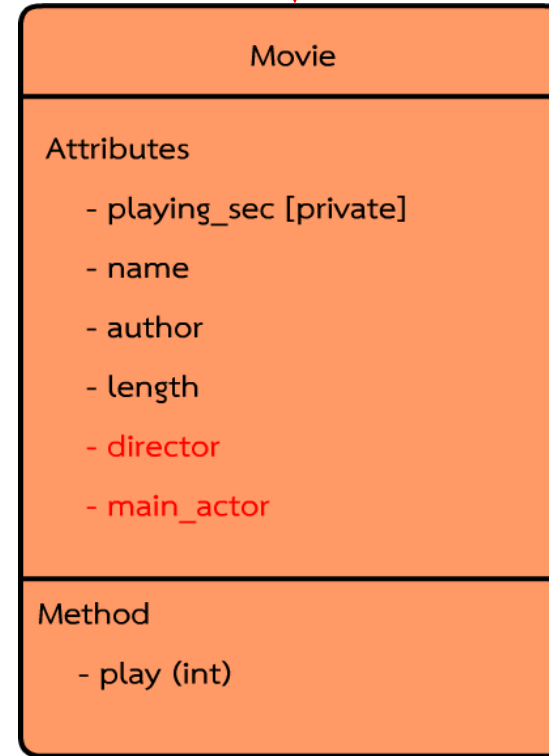


Parent Class  
Base Class

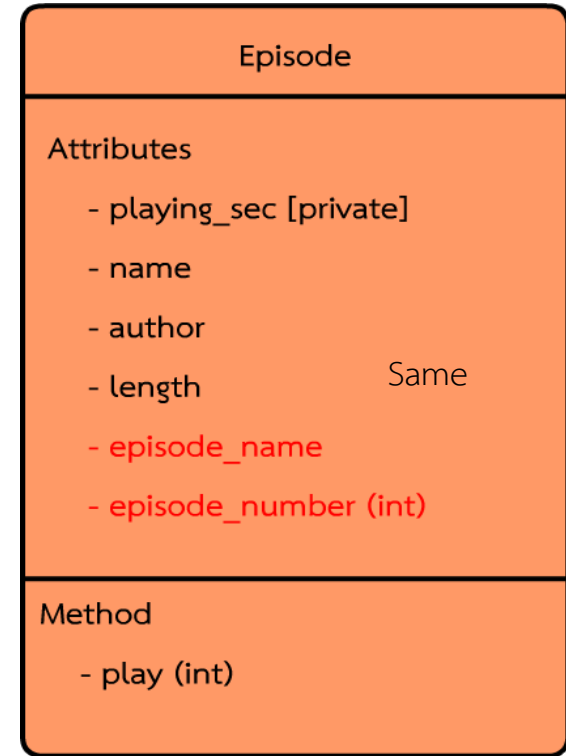
inherit (สืบทอด)



Child Class  
Derived Class



Child Class  
Derived Class



Child Class  
Derived Class

Constructor

---

QUIZ

# Constructor (default constructor)

---

```
72 class movie : public media{
73 public :
74     string director;
75     string main_actor;
76
77     movie(){
78         cout << "default constructor of [movie] class" << endl;
79     }
80 };
```

```
5 ~ class media{
6 public :
7
8 ~     media(){ // default constructor
9
10         cout << "default constructor of [media] class" << endl;
11
12         name = "unknow";
13         length = 0;
14         playing_sec = 0;
15     }
16
```

Code:

```
media m2;
```

Result:

default constructor of [media] class

# Constructor (default constructor)

---

```
72 class movie : public media{
73 public :
74     string director;
75     string main_actor;
76
77     movie(){
78         cout << "default constructor of [movie] class" << endl;
79     }
80 };
81
```

```
5 ~ class media{
6 public :
7
8 ~     media(){ // default constructor
9
10         cout << "default constructor of [media] class" << endl;
11
12         name = "unknow";
13         length = 0;
14         playing_sec = 0;
15     }
16
```

Code:

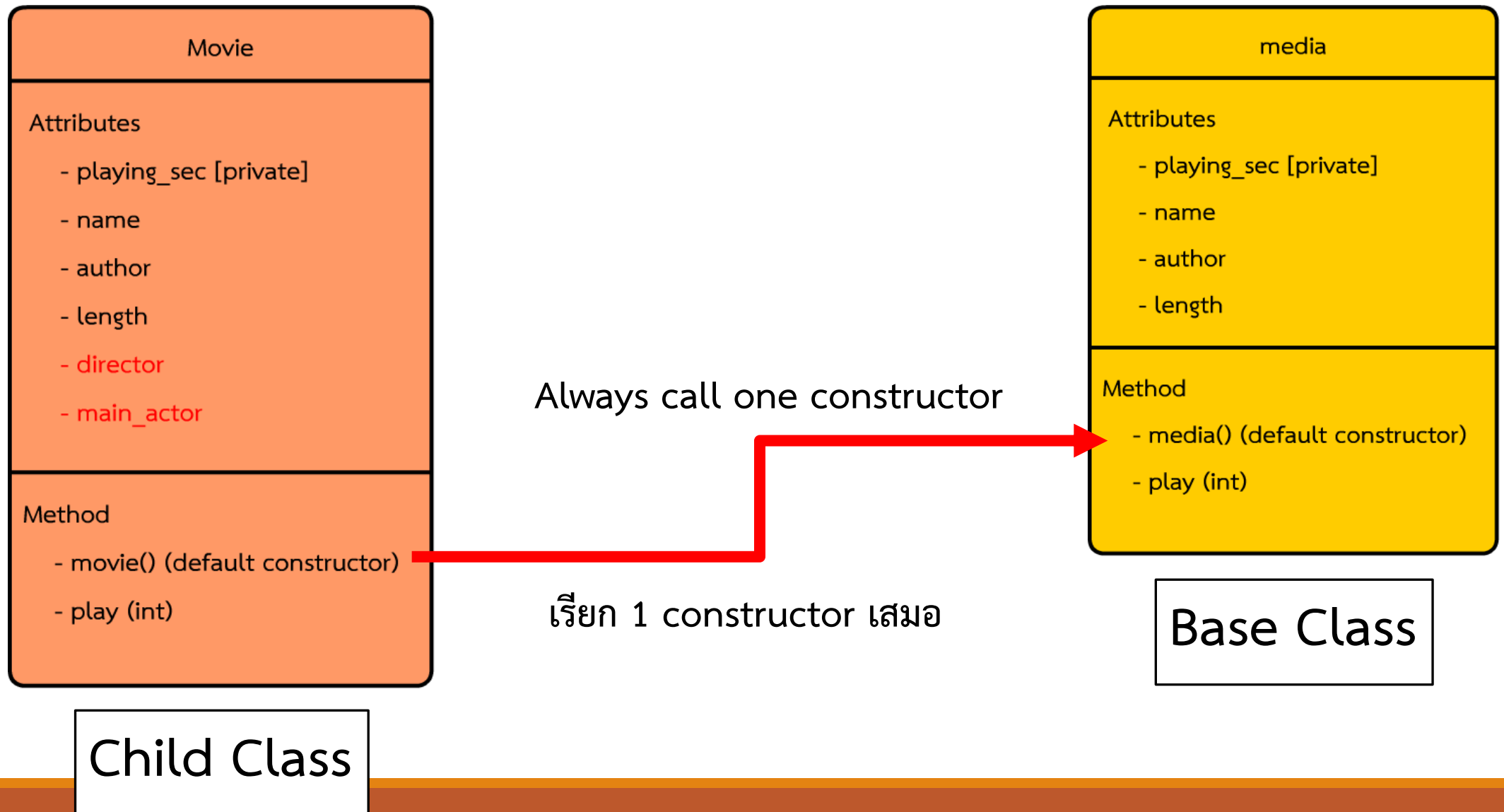
```
movie m1;
```

Result:

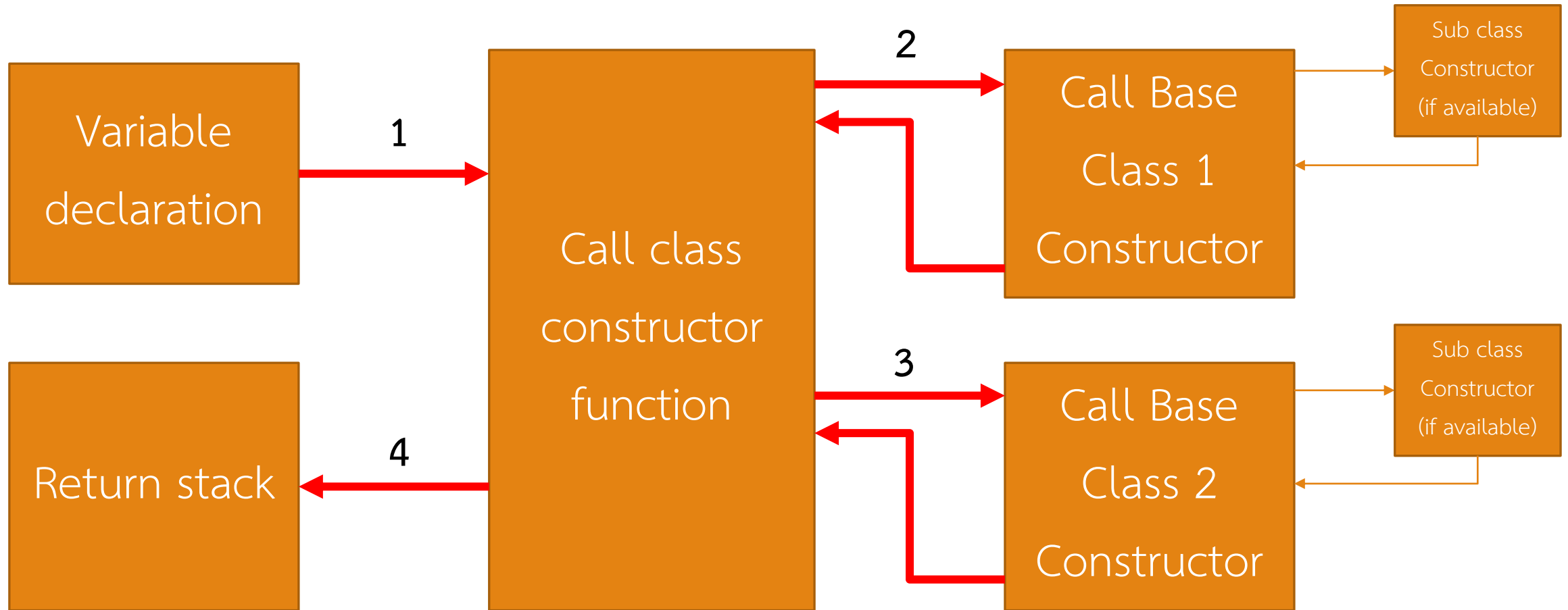
default constructor of [media] class

default constructor of [movie] class

# Constructor (default constructor)



# Constructor



# Constructor

---

```
song s1("Som San", "sek loso", 314, "LOSO");
```

```
song(string _name, string _author, int _length, string _brand){ // 4 parameter constructor  
  
    name = _name;  
    author = _author;  
    length = _length;  
    playing_sec = 0;  
    brand = _brand;  
  
}
```

But can we use existed media constructor? / แล้วสามารถใช้ constructor ของ media ได้หรือไม่



# Base constructor

---

```
media(string _name,string _author,int _length){ // 3 parameter constructor

    cout << "3 parameter constructor of [media] class" << endl;

    name = _name;
    author = _author;
    length = _length;
    playing_sec = 0;
}
```

Can we use this constructor in song class?

เราสามารถใช่ constructor นี้ใน class song ได้หรือไม่?

# Base constructor syntax

---

```
child_constructor : base_constructor1(p1,p2...), base_constructor2(p1,p2...) ){  
  
    cout << "4 parameter constructor of [song] class" << endl;  
    brand = _brand;  
}
```

# Base constructor

---

```
46 ~ class song : public media{
47     public :
48         string brand;
49
50 ~     song(){ // default constructor
51         cout << "default constructor of [song] class" << endl;
52     }
53
54 ~     song(string _name,string _author,int _length, string _brand) : media(_name,_author,_length){
55
56         cout << "4 parameter constructor of [song] class" << endl;
57         brand = _brand;
58
59 }
```

# Base constructor

---

```
46 ~class song : public media{
47     public :
48         string brand;
49
50 ~     song(){ // default constructor
51         cout << "default constructor of [song] class" << endl;
52     }
53
54 ~     song(string _name,string _author,int _length, string _brand) : media(_name,_author,_length){
55
56         cout << "4 parameter constructor of [song] class" << endl;
57         brand = _brand;
58
59     }
```

Code :

```
song s1("Som San","sek loso",314,"LOSO");

cout << s1.name << endl;
cout << s1.author << endl;
cout << s1.length << endl;
cout << s1.brand << endl;
```

Result :

3 parameter constructor of [media] class

4 parameter constructor of [song] class

Som San

sek loso

314

LOSO

# Base constructor another

---

```
movie(string _name,string _author,int _length, string _director, string _actor) : media(_name,_author,_length){  
    cout << "default constructor of [movie] class" << endl;  
    director = _director;  
    main_actor = _actor;  
}
```

```
episode(string _name,string _author,int _length, int ep_num, string ep_name) : media(_name,_author,_length){  
    cout << "default constructor of [episode] class" << endl;  
    episode_number = ep_num;  
    episode_name = ep_name;  
}
```

```

107     song s1("Som San","sek loso",314,"LOSO");
108
109     cout << s1.name << endl;
110     cout << s1.author << endl;
111     cout << s1.length << endl;
112     cout << s1.brand << endl;
113
114     song s2("Timemachine ","Pond Nipon",328,"Rap");
115     movie m1("The Disappearance of Haruhi Suzumiya","Nagaru Tanigawa",9707,"Tatsuya Ishihara","Haruhi");
116     movie m2("Avatar","James Cameron",9720,"James Cameron","Sam Worthington");
117     episode e1("Thi da Satan : ep 3 Earth Water Wind Fire","Kantana",3600,3,"Earth Water Wind Fire");
118     episode e2("Start-Up (2020) : ep 16 Scale up","Studio Dragon",4800,16,"Scale up");

```

Result :

- 3 parameter constructor of [media] class (107)
- 4 parameter constructor of [song] class (107)
- Som San (109)
- sek loso (110)
- 314 (111)
- LOSO (112)
- 3 parameter constructor of [media] class (114)
- 4 parameter constructor of [song] class (114)

- 3 parameter constructor of [media] class (115)
- 5 parameter constructor of [movie] class (115)
- 3 parameter constructor of [media] class (116)
- 5 parameter constructor of [movie] class (116)
- 3 parameter constructor of [media] class (117)
- 5 parameter constructor of [episode] class (117)
- 3 parameter constructor of [media] class (118)
- 5 parameter constructor of [episode] class (118)

# Constructor (from encapsulation)

---

- function ที่ถูกเรียกใช้ทุกครั้งที่มีการสร้าง object
  - ใช้เพื่อกำหนดค่าเริ่มต้นและตั้งค่าก่อนใช้ตัวแปร
  - สร้าง method ที่ชื่อเหมือน struct หรือ class
  - สามารถมี parameter ได้
  - Default constructor คือ constructor ที่ไม่รับ parameter หรือ มีแต่ default parameter
  - Default constructor จะถูก call เสมอหากไม่มี การ call constructor อื่น

# Quiz

---



# Inheritance Modifier

---

## private :

- accessible only in the same class , เข้าถึงได้เฉพาะใน class เดียวกัน

## protected :

- accessible in same class and Derived or child class
- เข้าถึงจาก class เดียวกัน และ class ที่สืบทอดไป (class ลูกๆ)

## public :

- accessible everywhere, เข้าถึงได้จากทุกที่

```

5 ~ class media{
6     private :
7         int playing_sec; // last played seconds in episode
8
9     protected :
10         string name;
11         string author;
12         int length; // media length in Seconds
13
14     public :
15 ~     void play(int time){ // play method
16         int remaining_time = length - playing_sec;
17         cout << "Playing " << name << " at[" << playing_s
18 ~         if(time > remaining_time){
19             playing_sec = 0;
20         }
21 ~         else{
22             playing_sec += time;
23         }
24     }
25
26 ~     int get_playing_sec(){
27         return playing_sec;
28     }
29
30 ~     void reset(){
31         playing_sec = 0;

```

- No one can edit playing sec directly
- ไม่มีใครสามารถเปลี่ยน playing sec โดยตรงได้
- Name ,author ,length สามารถเข้าถึงผ่านทาง class song ,movie ,episode ได้
- Name ,author ,length is accessible via song ,movie ,episode class
- Play(int) , get\_playing\_sec(), reset can be accessed from everywhere
- Play(int) , get\_playing\_sec(), reset สามารถเข้าถึงได้จากทุกที่

## C++ Syntax (from above slide)

---

```
class [child_class_name] : [modifier1] [base_class_name1] , [modifier2] [base_class_name2] ...{  
    // class component  
}
```

```
class movie : public media{  
public :  
    string director;  
    string main_actor;  
};
```

```
class episode : private media{  
public :  
    int episode_number;  
    string episode_name;  
};
```

# Common use Inheritance

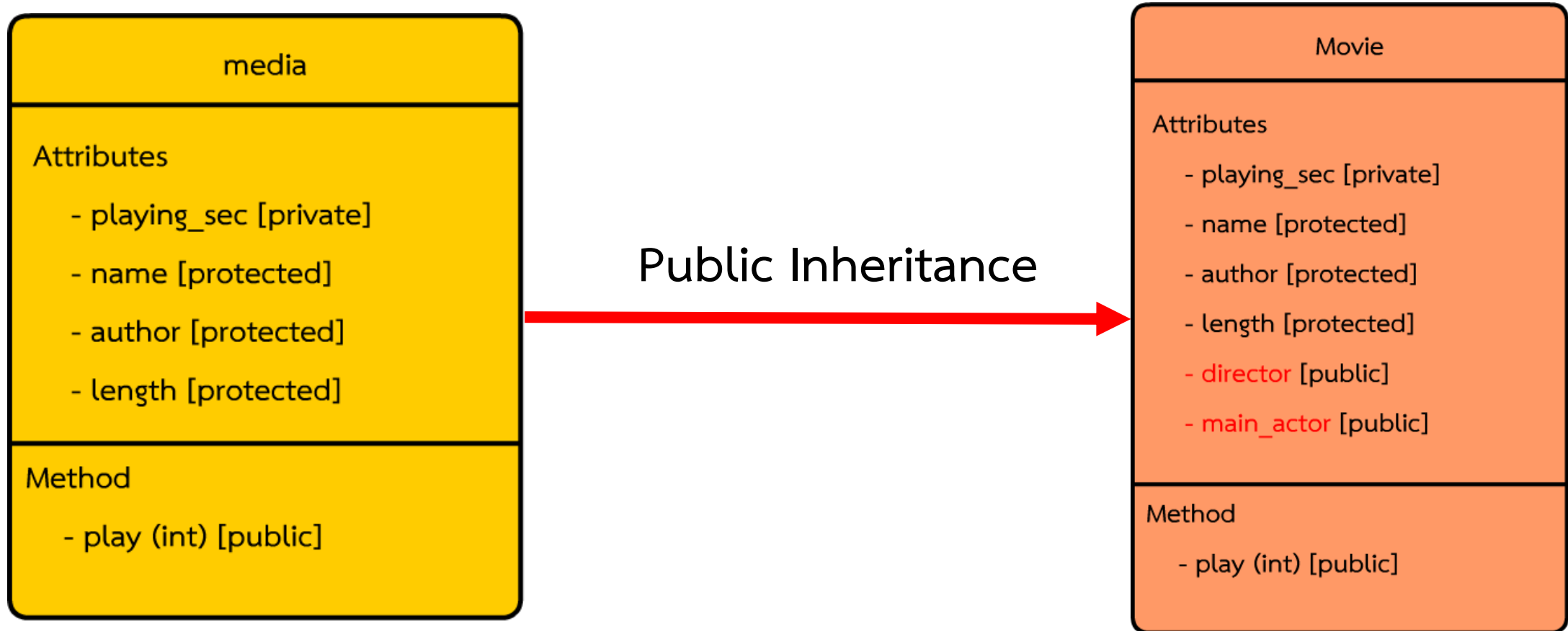
---

- **Public** inheritance is the most common usage
- การใช้งาน **Public** inheritance คือรูปแบบที่ใช้งานกันมากที่สุดและปกติที่สุด



```
class movie : public media{  
public :  
    string director;  
    string main_actor;  
};
```

# Public Inheritance



# Private and Protected inheritance

---

```
class movie : private media{  
public :  
    string director;  
    string main_actor;  
};
```

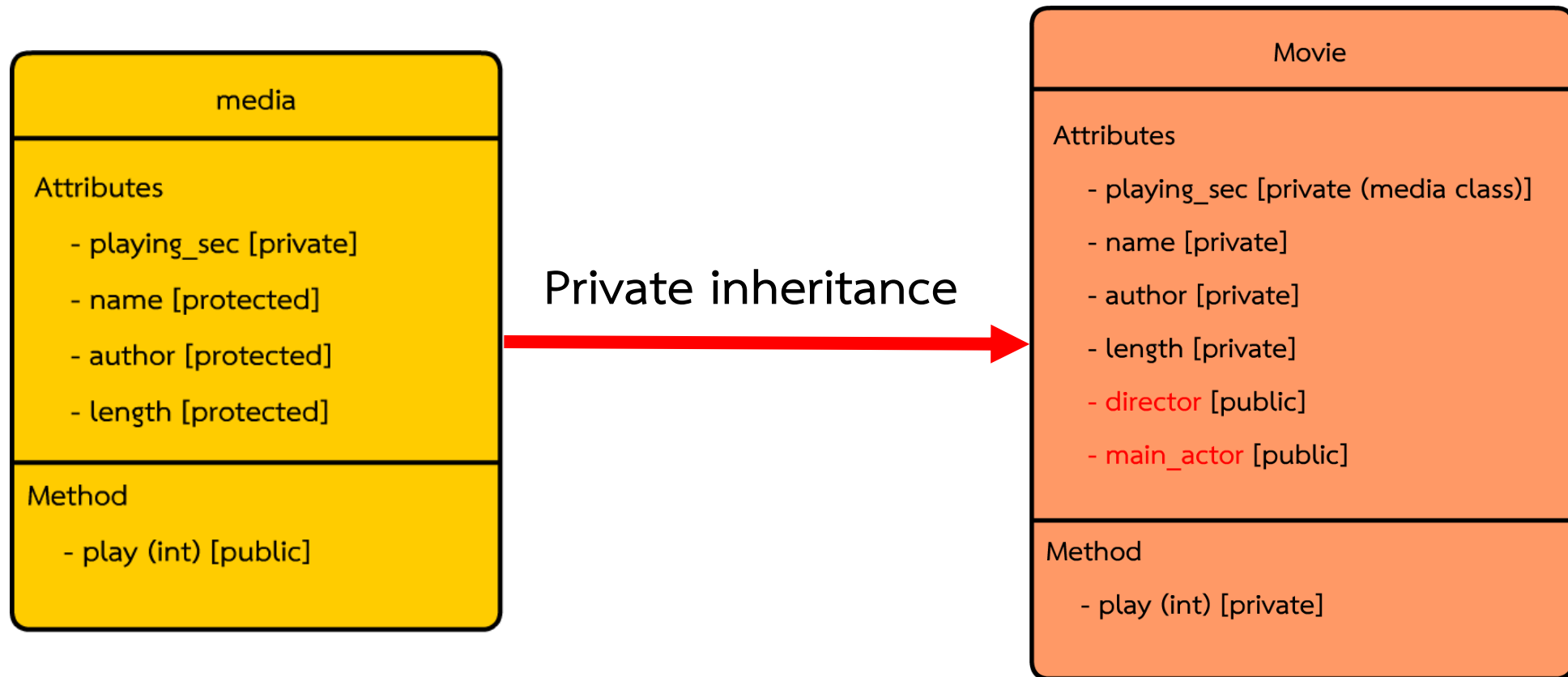
public      attribute **become** private of base class  
protected   attribute **become** private of child Class  
private      attribute **become** private of child Class

```
class movie : protected media{  
public :  
    string director;  
    string main_actor;  
};
```

public      attribute **become** protected  
protected   attribute **become** protected  
private      attribute **become** private of child Class

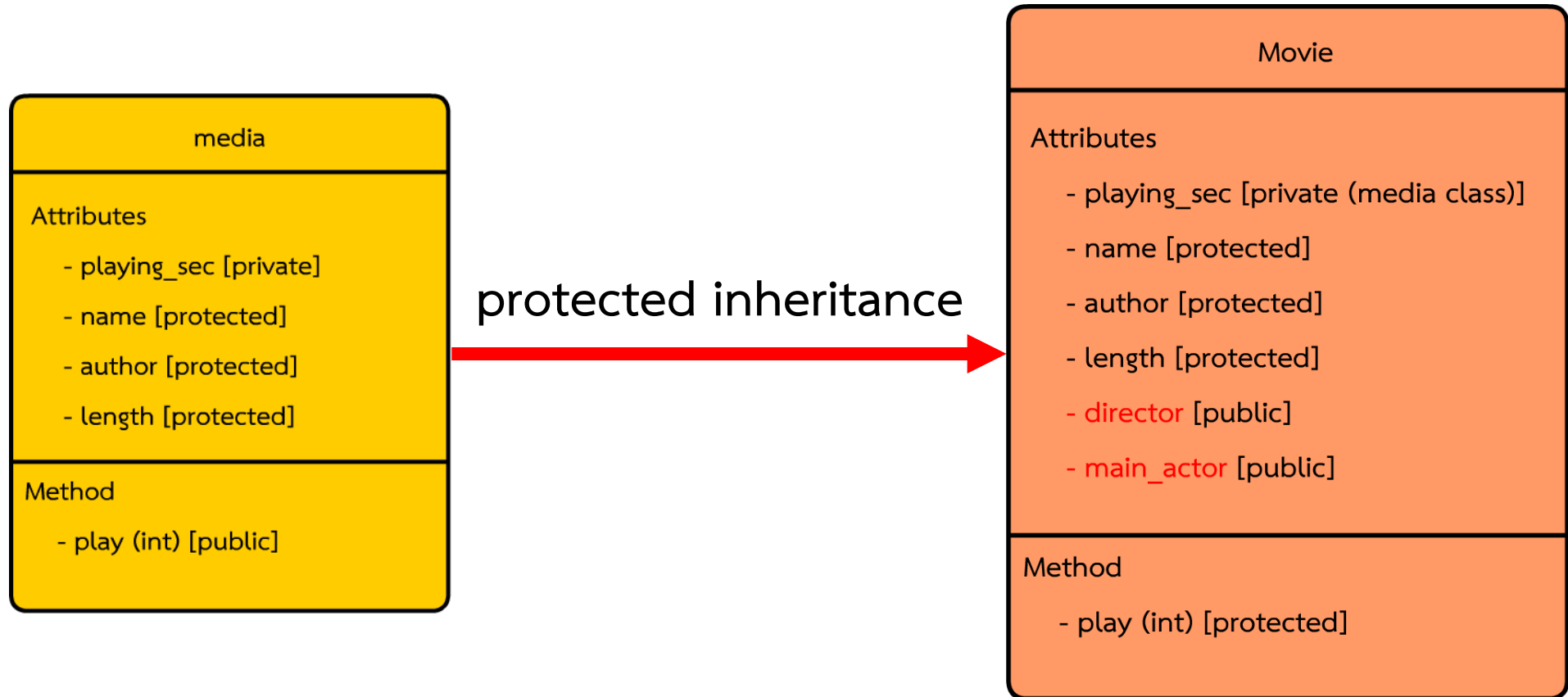
# Private inheritance

---



# Protected inheritance

---






# Modifier example

---

```
49 class song : public media{
50     private :
51         string brand;
52
53     public :
54         song(){ // default constructor
55         }
56
57         song(string _name,string _author,int _length, string _brand) : media(_name,_author,_length){ // 4 parameter
58             brand = _brand;
59         }
60
61         void print_song(){
62             cout << "now we playing " << name << " of " << brand << " at " << get_playing_sec() << " sec" << endl;
63         }
64     };
```

Code method :


```
void print_song(){  
    cout << "now we playing " << name << " of " << brand << " at " << get_playing_sec() << " sec";  
}
```



Media(private)



song(protected)



Media(public)

Code main :

```
s1.play(50);  
s1.print_song();
```

Result :

Playing Som San at[0] : [314 sec] remaining  
now we playing Som San of LOSO at 50sec

## Another example

---

Media(public)

```
void stop_song(){  
    reset();  
    cout << name << "has stopped" << endl;  
}
```

Media(private)

Code main :

```
s1.play(50);  
s1.print_song();  
s1.stop_song();  
s1.play(50);
```

Result :

Playing Som San at[0] : [314 sec] remaining  
now we playing Som San of LOSO at 50 sec  
Som San has stopped  
Playing Som San at[0] : [314 sec] remaining

# Move on from media

---

Base class :

```
5 ~ class vehicle{
6     protected :
7         int speed;
8         int wheel_count;
9
10        vehicle(int wheel){
11            wheel_count = wheel;
12            speed = 0;
13        }
14
15    public :
16        void print_wheel(){
17            cout << "this vehicle has [" << wheel_count << "] Wheel(s)" << endl;
18        }
19    };
20
```

```
class motorbike : public vehicle{
public :
    motorbike() : vehicle(2) {

    }

    void print(){
        cout << "motobike is using speed [" << speed << "]Kph" << endl;
    }
};
```

```
class airplane : public vehicle{
    int altitude;
public :

    void print(){
        cout << "airplane is at [" << altitude << "] ft above sealevel using speed [" <<
speed << "]Kph" << endl;
    }

    airplane(int wheel) : vehicle(wheel) {
        altitude = 0;
    }
};
```

```
class submarine : public vehicle{
    int depth;
public :

    void print(){
        cout << "submarine is at [" << depth << "] depth level using speed [" << speed << "]Kph" << endl;
    }

    submarine() : vehicle(0) {
        depth = 0;
    }
};
```

```
motorbike vaspasprint;  
airplane a380(22);  
submarine S26T;  
  
vaspasprint.print_wheel(); // this vehicle has [2] Wheel(s)  
a380.print_wheel(); // this vehicle has [22] Wheel(s)  
S26T.print_wheel(); // this vehicle has [0] Wheel(s)  
  
vaspasprint.print(); // motobike is using speed [0]Kph  
a380.print(); // airplane is at [0] ft above sealevel using speed [0]Kph  
S26T.print(); // submarine is at [0] depth level using speed [0]Kph
```

Result :

this vehicle has [2] Wheel(s)

this vehicle has [22] Wheel(s)

this vehicle has [0] Wheel(s)

motobike is using speed [0]Kph

airplane is at [0] ft above sealevel using speed [0]Kph

submarine is at [0] depth level using speed [0]Kph

# LAB

---



## ข้อกำหนด

---

- ห้ามเปลี่ยน modifier ของไฟล์ LAB
  - ห้ามเปลี่ยน main
  - นอนนั้นทำได้ทุกอย่าง เขียนเพิ่มเติมที่ไหนของไฟล์ก็ได้
- Global variable , เพิ่ม constructor ได้