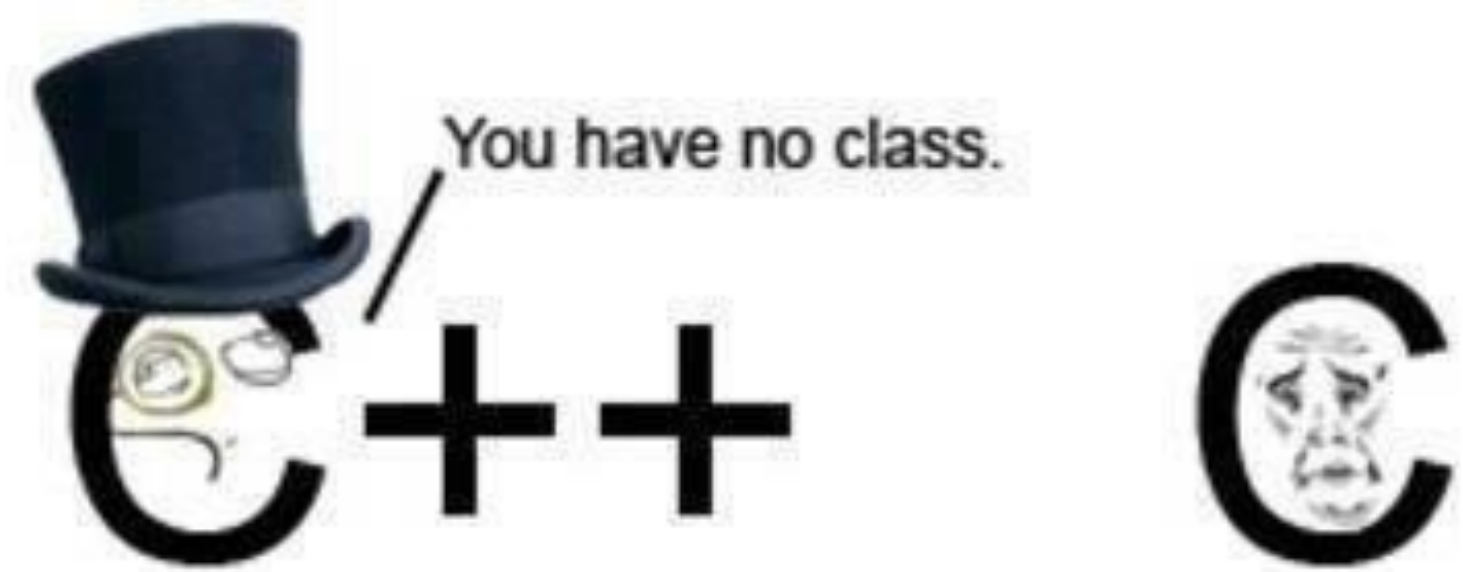
OOP & data struct

2. Class concept

BY SOMSIN THONGKRAIRAT

DR.ATTASIT LASAKUL EDITION


You have no class.
C++
C

What is Object-oriented programming (OOP)

Programming paradigm using “OBJECT” concept

รูปแบบหนึ่งของการเขียนโปรแกรมที่มองทุกอย่างเป็น “OBJECT”

Most popular of Programming agreement (2022)

เป็นรูปแบบที่ใช้กันมากที่สุด (2565)

Data type ยังคงเหมือนเดิม (คล้ายภาษา C)

C++

int – integer (จำนวนเต็ม)

float – floating point (ทศนิยม)

string text = “Hello”;

C

```
int number = 50;
float  height = 75.6;
char name[3]={‘K’,’L’,’\0’};
```

Example

```
int     my_integer          = 10;
int     my_integer2         = 20;
float   my_floating_point   = 3.14159;.
```

Variable type? ชนิดของตัวแปร?

Variable name? ชื่อของตัวแปร?

Variable value? ค่าของตัวแปร?

ในภาษา C เรามีข้อมูลแบบ struct ที่สามารถกำหนดข้อมูลชนิดอื่นไว้ในตัวมันได้

ในภาษา C++ ก็มีข้อมูลแบล struct เหมือนกันแต่จะมีความสามารถมากกว่าที่ชัดเจนคือ คุณสมบัติของการ สืบทอดได้ เป็นต้น

และใน C++ ข้อมูลชนิด Struct จะเป็นเหมือน class 98% ต่างกันเพียงมีการกำหนดค่าตัวแปรหรือฟังก์ชั่น ภายในอัตโนมัติเป็นแบบ.....

ของ struct จะเป็น => public

ของ Class จะเป็น => private

Type Declaration

กำหนด Struct เป็นข้อมูลชนิดใหม่ไว้ใช้งาน (เป็นเหมือนแม่แบบ)

```
// struct declaration
struct anime{
    string full_name;
    string author;
    int total_episode;
    int length_per_episode;
};
```

Variable Declaration

ใน main( ) เรากำหนดตัวแปร a1, a2, a3, a4 ให้เป็นข้อมูลชนิดใหม่นั้นและใช้งาน (เป็นการสร้าง Object นั่นเอง)

```
int main()
{
    anime a1, a2;
    a1.full_name = "The Melancholy of Haruhi Suzumiya";
    a1.author = "Nagaru Tanigawa";
    a1.total_episode = 2;
    a1.length_per_episode = 1200;
    a1.playing_episode = 1;
    a1.playing_sec = 0;
```

Object Concept สามารถสรุปเบื้องต้นได้เป็น

- Class คือ blueprint หรือต้นแบบในการสร้าง object (data type)
- Object (instance) คือ ตัวแทนที่ถูกสร้างขึ้นมาจาก class (ตัวแปร)
- Attributes คือ ข้อมูลหรือตัวแปรใน struct (class)
- Method คือ การทำงานของ object (คำสั่ง หรือ function) ***

Example โปรแกรม 2_1.....CPP

<u>เป็นโปรแกรมในการเล่นภาพยนตร์</u> โดยจะสร้างคลาส anime เพื่อเอาไว้สร้าง object-ของภาพยนต์แต่ละเรื่องโดยข้างในก็จะมีการจัดเก็บข้อมูลดังนี้

(1) ชื่อภาพยนตร์ full_name **(2) ชื่อผู้แต่ง author**

(3) จำนวนของตอน total_episode

(4) ความยาวของตอนแต่ละตอน (เท่ากัน) total_per_episode

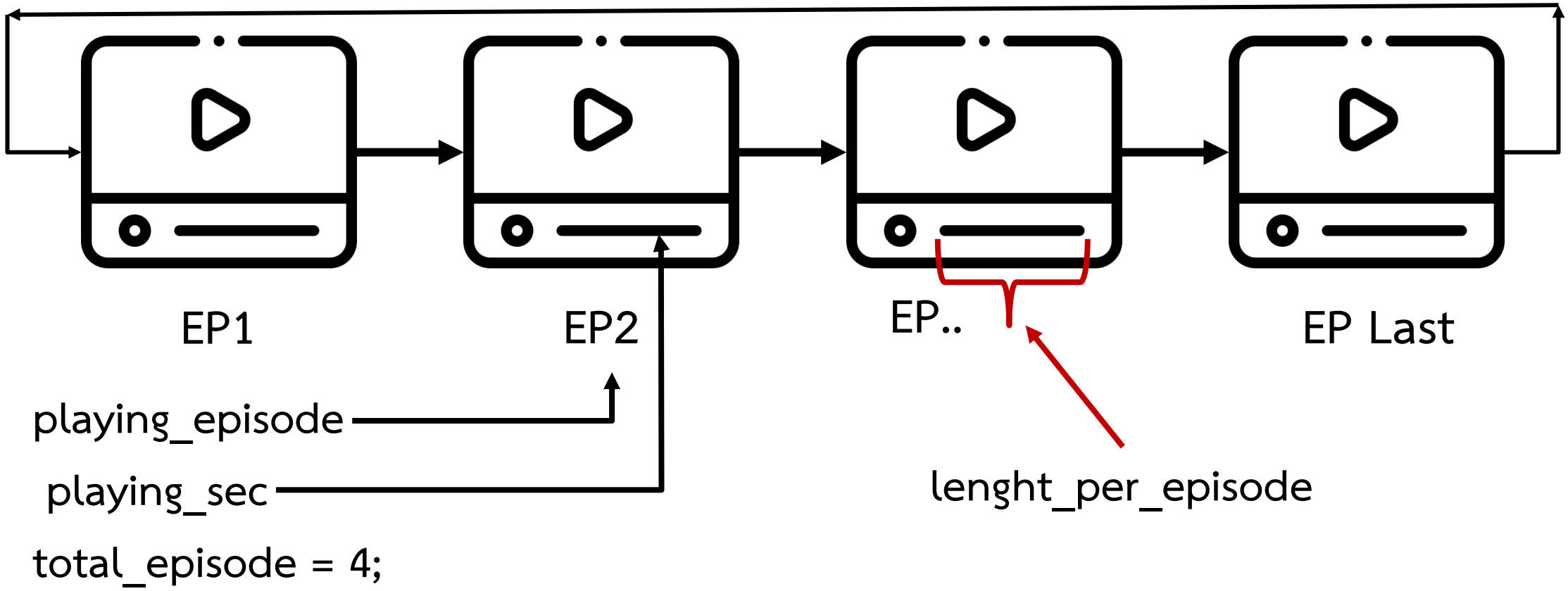
(5) หมายเลขตอนที่เล่น playing_episode

(6) เวลารวมที่เล่นของแต่ละตอน playing_sec;

สร้าง ฟังก์ชัน play( ) ภาพยนตร์ เพื่อเล่นภาพยนต์คือ

- สั่งให้เล่นภาพยนตร์ ได้โดยสามารถกำหนดจำนวนเวลาเล่นได้
- ภาพยนตร์สั่งเล่นแล้วสามารถเก็บสถานะเดิมเพื่อมาเล่นต่อเนื่องได้
- สั่งเล่นเกินเวลาของแต่ละตอน ก็เล่นจบที่เวลาเหลือของตอนนั้นๆ แล้วเมื่อสั่งเล่นใหม่ก็จะต่อเนื่องในตอนต่อไป
- หากสั่งเล่นจบทุกตอนแล้ว เมื่อสั่งเล่นใหม่ก็จะวนไปเริ่มเล่นตอนที่ 1

การทำงานของ function (method) ที่ใช้เล่นคือ void play(Object, time)



คลาส (struct) anime ในโปรแกรม 2_1.cpp เรามีโครงสร้างคลาสดังนี้

Struct/class

methods

main

```
#include <iostream>
using namespace std;

struct anima
  {
      string fullname;
      ..................
  }

void play(anine *a, int time)
{
    int remain_time = .....
          ......................
}

int main( )
{
    anime a1, a2 ,....
     play(&a1,500);
      ..................
}
```

```
struct anime
{
    string full_name;
    string author;
    int total_episode;
    int length_per_episode; // Average running time in Seconds

    int playing_episode; // last played episode
    int playing_sec;     // last played seconds in episode
};
```

เรานำมาสร้างเป็น Object a1,a2 เพื่อนำไปใช้กับ play( ) ดังนี้

```
int main()
{
    anime a1, a2;
    a1.full_name = "The Melancholy of Haruhi Suzumiya";
    a1.author = "Nagaru Tanigawa";
    a1.total_episode = 2;
    a1.length_per_episode = 1200;
    a1.playing_episode = 1;
    a1.playing_sec = 0;
```

ใน main( ) เราสั่งให้เล่นภาพยนตร์ a1 ตามนี้ (กำหนดให้ ฟังก์ชั่น play รับ Object ภาพยนต์แบบ pointer)

main( )

```
cout << "1. play a1 time 200 sec" << endl;
play(&a1, 200);
cout << "2. play a1 time 500 sec" << endl;
play(&a1, 500);
cout << "3. play a1 time 999999 sec" << endl;
play(&a1, 99999);
cout << "4. play a1 time 700 sec" << endl;
play(&a1, 700);
cout << "5. play a1 time 700 sec" << endl;
play(&a1, 700);
cout << "6. play a1 time 700 sec" << endl;
play(&a1, 700);
```

เมื่อสั่งเล่น a1 ต่อไปดังนี้

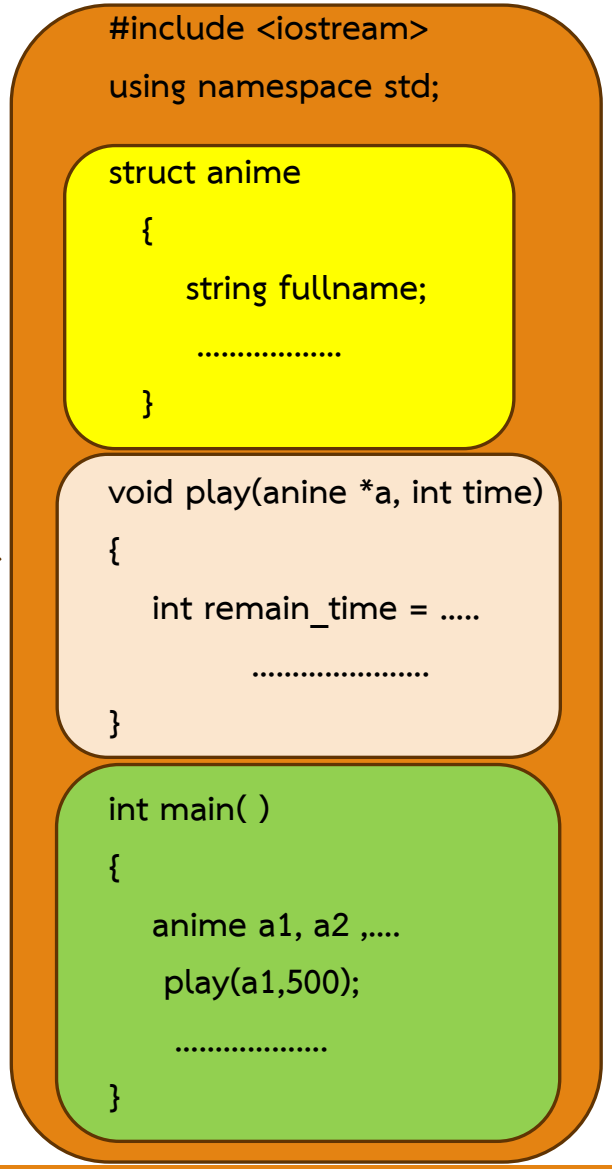
```
cout << "1. play a1 time 200 sec" << endl;
play(&a1, 200);
cout << "2. play a1 time 500 sec" << endl;
play(&a1, 500);
cout << "3. play a1 time 999999 sec" << endl;
play(&a1, 99999);
cout << "4. play a1 time 700 sec" << endl;
play(&a1, 700);
cout << "5. play a1 time 700 sec" << endl;
play(&a1, 700);
cout << "6. play a1 time 700 sec" << endl;
play(&a1, 700);
```

```
1. play a1 time 200 sec
 playing => [The Melancholy of Haruhi Suzumiya] Author => [Nagaru Tanigawa]
 Episode => [EP.1] playing time =  [200 sec]
2. play a1 time 500 sec
 playing => [The Melancholy of Haruhi Suzumiya] Author => [Nagaru Tanigawa]
 Episode => [EP.1] playing time =  [700 sec]
3. play a1 time 999999 sec
 playing => [The Melancholy of Haruhi Suzumiya] Author => [Nagaru Tanigawa]
 Episode => [EP.1] playing time =  [1200 sec]
  ---> full Episode <---
4. play a1 time 700 sec
 playing => [The Melancholy of Haruhi Suzumiya] Author => [Nagaru Tanigawa]
 Episode => [EP.2] playing time =  [700 sec]
5. play a1 time 700 sec
 playing => [The Melancholy of Haruhi Suzumiya] Author => [Nagaru Tanigawa]
 Episode => [EP.2] playing time =  [1200 sec]
  ---> full Episode <---
  ---> All Episodes completed. Restarting from Episode 1 <---
6. play a1 time 700 sec
 playing => [The Melancholy of Haruhi Suzumiya] Author => [Nagaru Tanigawa]
 Episode => [EP.1] playing time =  [700 sec]
```

ฟังก์ชั่นหรือ Method play(...) (แบบใส่ไว้นอก struct)

```
void play(anime *a, int time)
{
    a->playing_sec += time;
    if (a->playing_sec >= a->length_per_episode)
        {a->playing_sec = a->length_per_episode;}
    int remaining_time = a->length_per_episode - a->playing_sec;
    if (remaining_time < 0)
        {remaining_time = 0;}
    cout << " playing => [" << a->full_name << "]" << " Author => [" << a->author << "]\n";
    cout << " Episode => [EP." << a->playing_episode;
    cout << "] playing time = " << " [" << a->playing_sec << " sec]\n";
    if (a->playing_sec >= a->length_per_episode)
      { a->playing_sec = 0;
        a->playing_episode++;
        cout << "  ---> full Episode <---\n";
        if (a->playing_episode > a->total_episode)
        {cout << "  ---> All Episodes completed. Restarting from Episode 1 <---\n";
         a->playing_episode = 1;}
    }
}
```

โครงร่างโปรแกรม 2_1..CPP

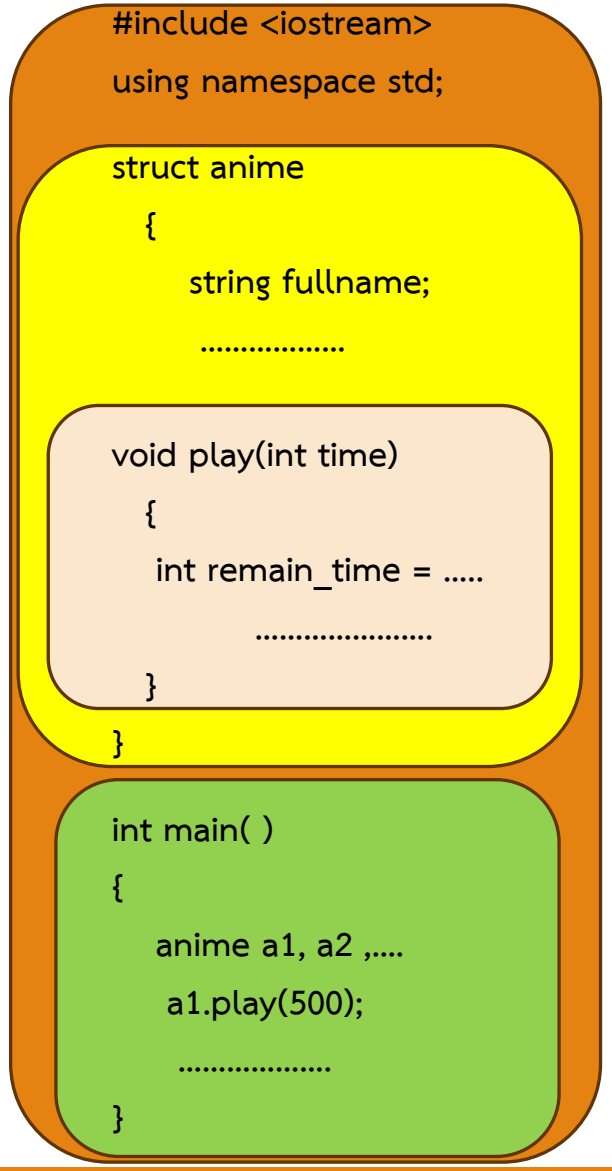


struct

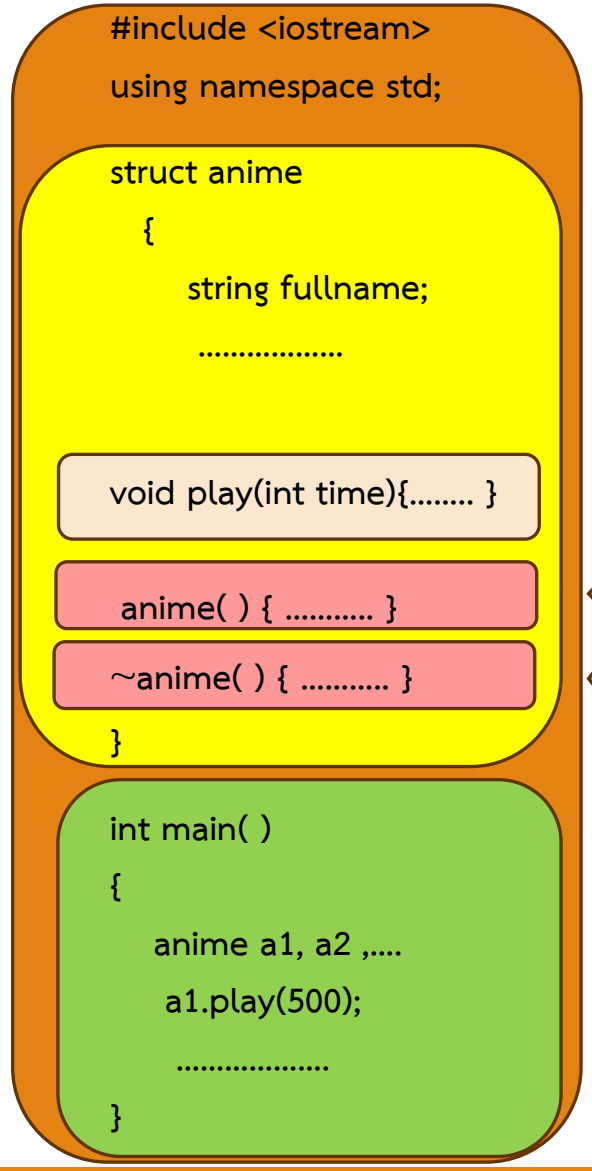
methods

main

โครงร่างโปรแกรม 2_2..CPP



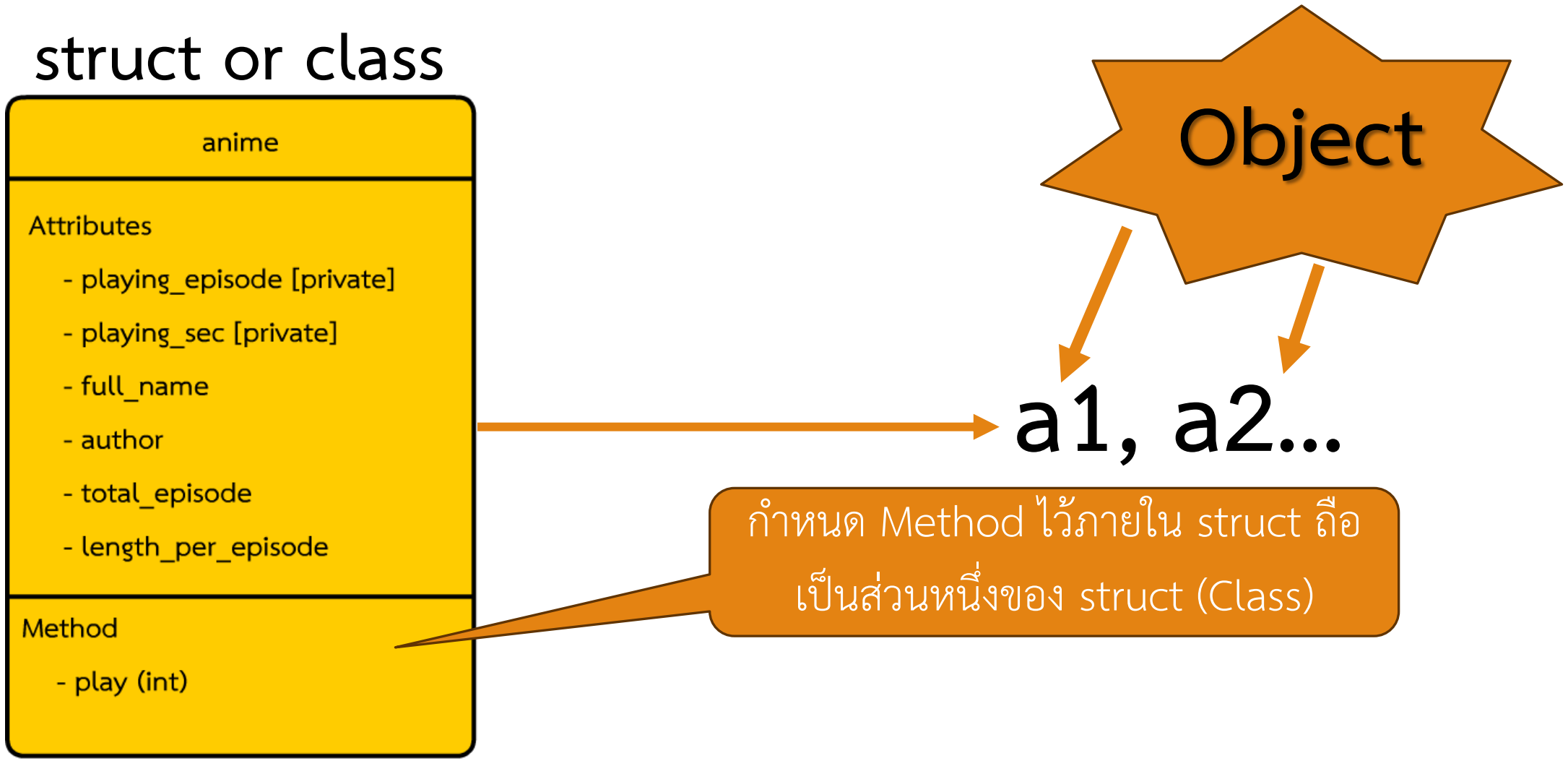
โครงร่างโปรแกรม 2_3 และ 2_4..CPP



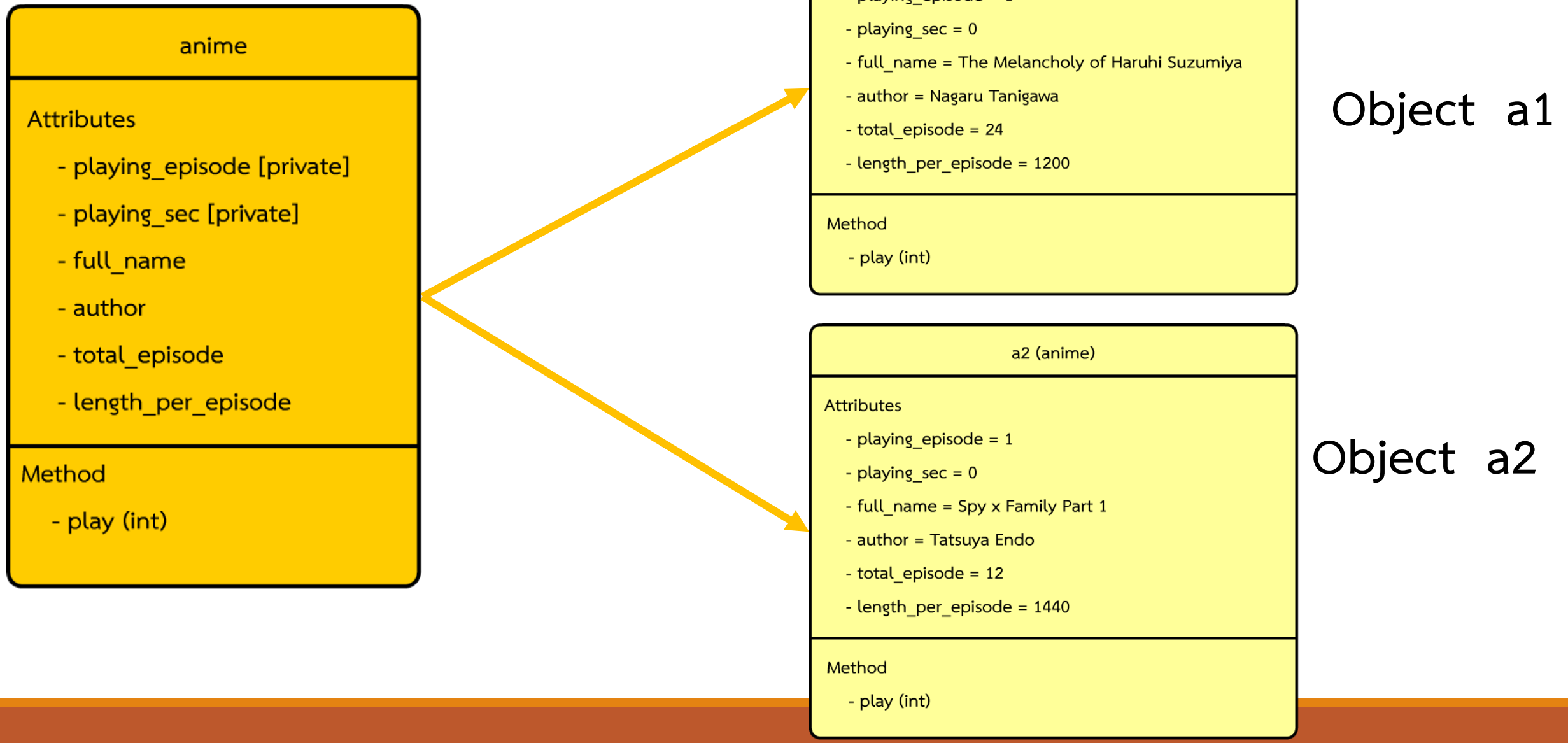
constructor

deconstructor

โปรแกรม 2_2...CPP รวม (methodไว้ด้วย)



Struct or Class



Let’s make it ให้มีทุกอย่างอยู่ภายใน struct (class) ตัวเอง

โปรแกรม 2_2...CPP

```
struct anime
{
    string full_name;
    string author;
    int total_episode;
    int length_per_episode; // Average running time in Seconds

    int playing_episode; // last played episode
    int playing_sec;     // last played seconds in episode

    void play(int time)
     {  playing_sec += time;
        if (playing_sec >= length_per_episode)
            {playing_sec = length_per_episode;}
        int remaining_time = length_per_episode - playing_sec;
        if (remaining_time < 0)
            {remaining_time = 0;}
        cout << " playing => [" << full_name << "]" << " Author => [" << author << "]\n";
        cout << " Episode => [EP." << playing_episode;
        cout << "] playing time = " << " [" << playing_sec << " sec]\n";
        if (playing_sec >= length_per_episode)
            {playing_sec = 0;
            playing_episode++;
            cout << "  ---> full Episode <---\n";
            if (playing_episode > total_episode)
                {cout << "  ---> All Episodes completed. Restarting from Episode 1 <---\n";
                playing_episode = 1;}}}
};
```

Attributes

Method

สร้าง Object a1, a2 และกำหนดค่าเริ่มต้น

```
// variable declaration
anime a1,a2;
a1.full_name = "The Melancholy of Haruhi Suzumiya";
a1.author = "Nagaru Tanigawa";
a1.total_episode = 24;
a1.length_per_episode = 1200;
a1.playing_episode = 1;
a1.playing_sec = 0;

a2.full_name = "Spy x Family Part 1";
a2.author = "Tatsuya Endo";
a2.total_episode = 12;
a2.length_per_episode = 1440;
a2.playing_episode = 1;
a2.playing_sec = 0;
```

ก็ใช้งานได้ปกติ

```
a1.play(500);
a1.play(500);
a1.play(99999);
a1.play(99999);
a2.play(100);
a2.play(100);
a2.play(100);
a2.play(99999);
a1.play(500);
a1.play(99999);
```

ความแตกต่างของฟังก์ชั่น play() ที่อยู่ภายในและภายนอก struct

ส่งค่าที่อยู่ของตัวแปร a และค่าเวลา

```
void play(anime *a, int time)
{
    a->playing_sec += time;
    if (a->playing_sec >= a->length_per_episode)
        {a->playing_sec = a->length_per_episode;}
    int remaining_time = a->length_per_episode - a->playing_sec;
    if (remaining_time < 0)
        {remaining_time = 0;}
    cout << " playing => [" << a->full_name << "]" << " Author => [" << a->author << "]\n";
    cout << " Episode => [EP." << a->playing_episode;
    cout << "] playing time = " << " [" << a->playing_sec << " sec]\n";
    if (a->playing_sec >= a->length_per_episode)
      { a->playing_sec = 0;
        a->playing_episode++;
        cout << "  ---> full Episode <---\n";
        if (a->playing_episode > a->total_episode)
        {cout << "  ---> All Episodes completed. Restarting from Episode 1 <---\n";
         a->playing_episode = 1;}
    }
}
```

Procedural (function)

ส่งเฉพาะค่าเวลาเท่านั้น

```
void play(int time)
{  playing_sec += time;
    if (playing_sec >= length_per_episode)
        {playing_sec = length_per_episode;}
    int remaining_time = length_per_episode - playing_sec;
    if (remaining_time < 0)
        {remaining_time = 0;}
    cout << " playing => [" << full_name << "]" << " Author => [" << author << "]\n";
    cout << " Episode => [EP." << playing_episode;
    cout << "] playing time = " << " [" << playing_sec << " sec]\n";
    if (playing_sec >= length_per_episode)
        {playing_sec = 0;
        playing_episode++;
        cout << "  ---> full Episode <---\n";
        if (playing_episode > total_episode)
            {cout << "  ---> All Episodes completed. Restarting from Episode 1 <---\n";
            playing_episode = 1;}}}
```

OOP (method)

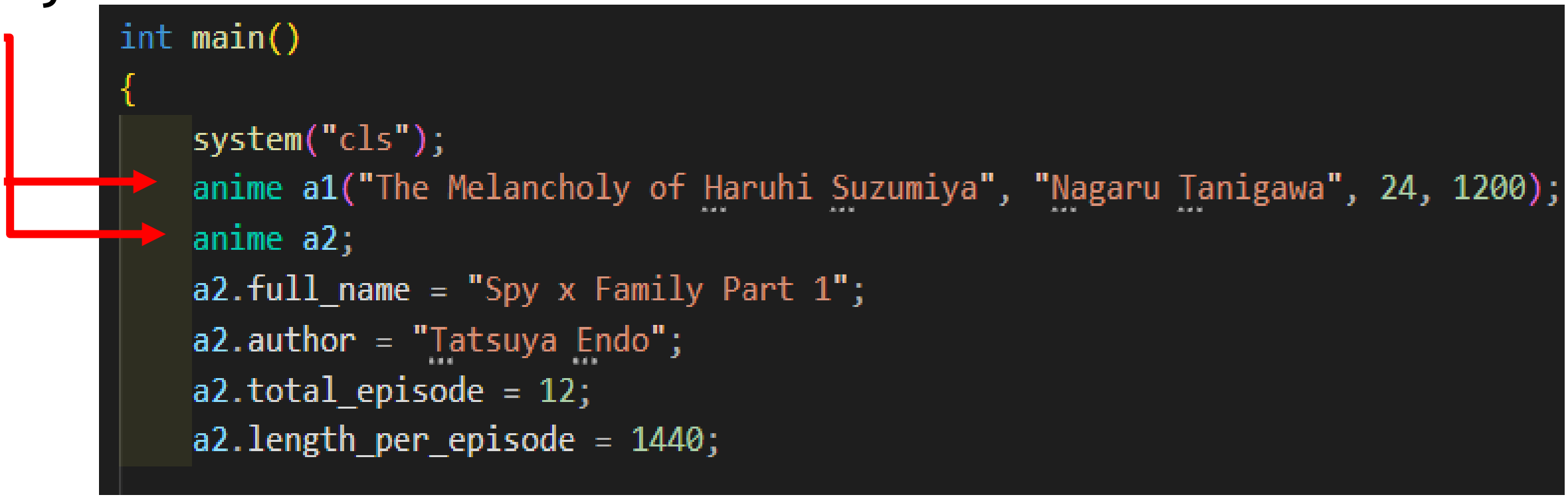
ได้ผลลัพธ์เท่ากันแต่เขียนง่ายกว่า Same result

```
play(&a1, 500);
play(&a1, 500);
play(&a1, 900);
play(&a1, 100);
```

```
a1.play(500);
a1.play(500);
a1.play(900);
a1.play(100);
```

เราสามารถกำหนดค่าเริ่มต้นให้ Object ตัวแปรได้โดยใช้ Constructor (โปรแกรมที่ 2_3.CPP)

Object

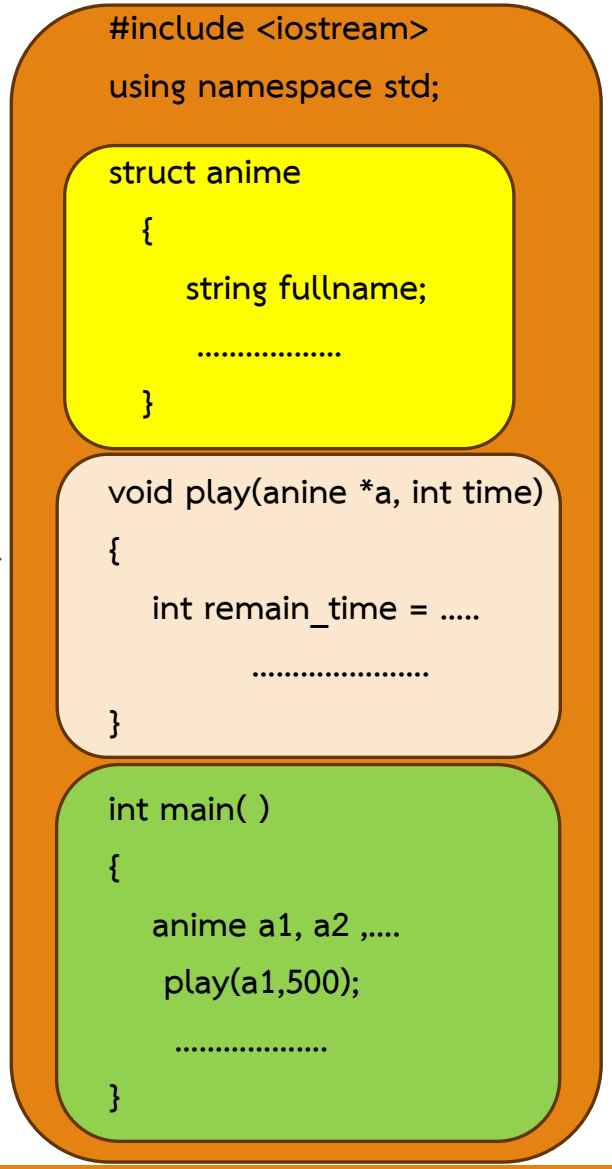


```
int main()
{
    system("cls");
    anime a1("The Melancholy of Haruhi Suzumiya", "Nagaru Tanigawa", 24, 1200);
    anime a2;
    a2.full_name = "Spy x Family Part 1";
    a2.author = "Tatsuya Endo";
    a2.total_episode = 12;
    a2.length_per_episode = 1440;
```

Constructor คือ.....

- **function ที่ถูกเรียกใช้ทุกครั้งที่มีการสร้าง object**
  - ใช้เพื่อการกำหนดค่าเริ่มต้นและตั้งค่าก่อนใช้ตัวแปร
  - เป็น method ที่ชื่อเหมือน struct หรือ class
  - สามารถมี parameter ได้และไม่มีการส่งค่ากลับ
  - Default constructor คือ constructor ที่ไม่รับ parameter หรือ มีแต่ default parameter
  - Default constructor จะถูก call เสมอหากไม่มี การ call constructor อื่น

โครงร่างโปรแกรม 2_1..CPP

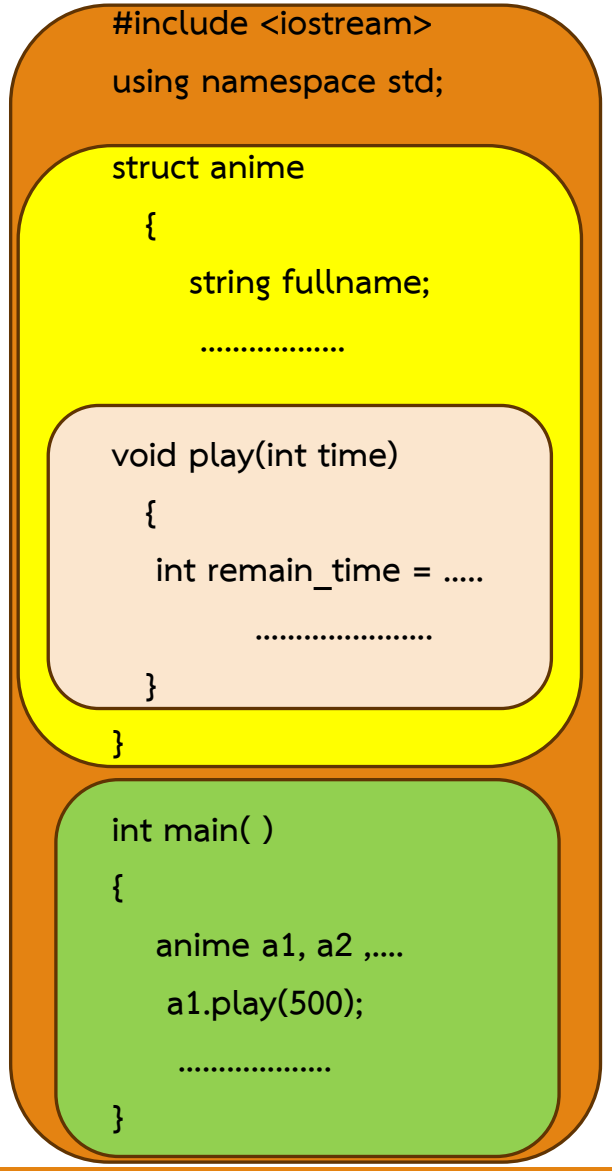


struct

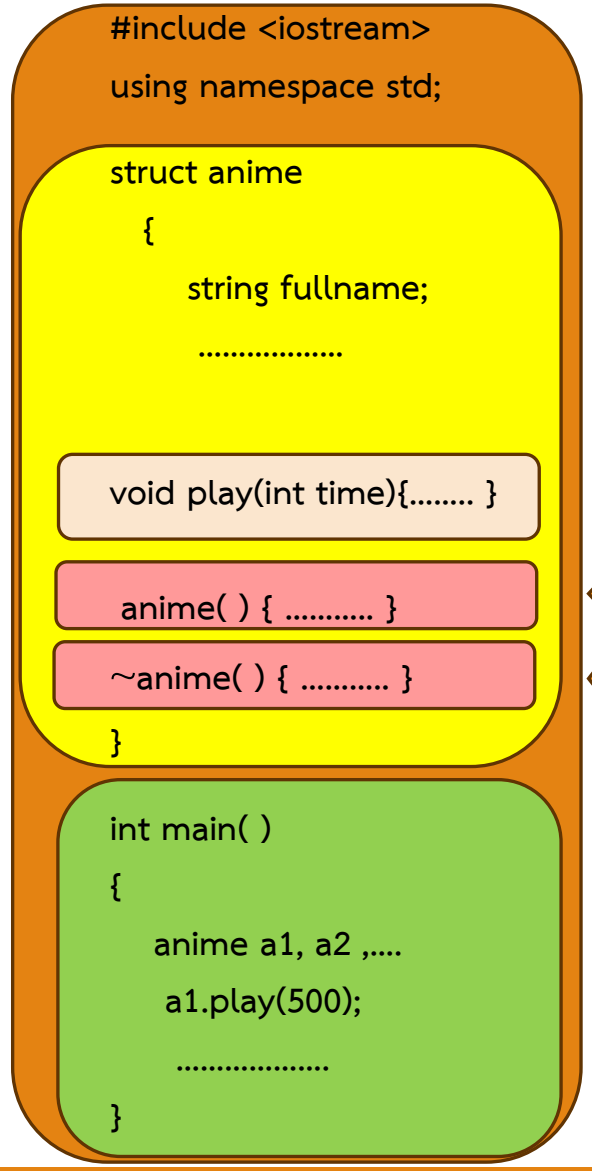
methods

main

โครงร่างโปรแกรม 2_2..CPP



โครงร่างโปรแกรม 2_3 และ 2_4..CPP



constructor

deconstructor

ตัวอย่าง Constructor แบบไม่มีการรับค่าใดๆ โปรแกรม 2_3...CPP

```cpp
struct anime{
    string full_name;
    string author;
    int total_episode;
    int length_per_episode; // Average running time in Seconds

    int playing_episode; // last played episode
    int playing_sec; // last played seconds in episode

anime(){
    cout << "this is default constructor" << endl;

    playing_episode = 1;
    playing_sec = 0;
}
```

จะเป็นแบบไม่มีการรับค่า (Default constructor) และแบบมีการรับค่าสามารถกำหนดไว้หลายๆแบบได้ใน Struct หรือ Class เดียวกันนั้น

```
41      anime(){
42          cout << "this is default constructor" << endl;
43
44          playing_episode = 1;
45          playing_sec = 0;
46      }
47
48      anime(string _name,string _author,int _ep,int length){
49          cout << "this is constructor for " << _name << endl;
50
51          full_name = _name;
52          author = _author;
53          total_episode = _ep;
54          length_per_episode = length;
55
56          playing_episode = 1;
57          playing_sec = 0;
58      }
59
```

Default constructor

Constructor with argument

```cpp
int main()
{
    system("cls");
    anime a1("The Melancholy of Haruhi Suzumiya", "Nagaru Tanigawa", 24, 1200);
    anime a2;
    a2.full_name = "Spy x Family Part 1";
    a2.author = "Tatsuya Endo";
    a2.total_episode = 12;
    a2.length_per_episode = 1440;
```

Call constructor with parameter

Call default constructor

Result

```
this is constructor for The Melancholy of Haruhi Suzumiya
this is default constructor
```

การกำหนดรูปแบบการเข้าถึงของตัวแปร (Variable modifiers)

- เพื่อจำกัดการเข้าถึง

(ใช้ได้ทั้งกับ methods and attributes)

- มี 3 รูปแบบ

public protected private

พิจารณากรณีมีการเปลี่ยนค่าตัวแปรข้างล่าง

```
a1.playing_episode = -20;
a1.play(500);
a1.play(500);
```

Result

```
playing => [The Melancholy of Haruhi Suzumiya] Author => [Nagaru Tanigawa]
Episode => [EP.-20] playing time =  [500 sec]
playing => [The Melancholy of Haruhi Suzumiya] Author => [Nagaru Tanigawa]
Episode => [EP.-20] playing time =  [1000 sec]
```

เราสามารถป้องกันโดยการ (1) กำหนดชนิดการเข้าถึงตัวแปรเป็น private และ (2)สร้าง method เพื่อการเปลี่ยนค่าในขอบเขตที่ถูกต้องใส่ใน struct ไว้ได้

(1)

```
struct anime
{
private:
    int playing_episode; // last
public:
    string full_name;
    string author;
    int total_episode;
    int length_per_episode; // A
```

(2)

```
void select_episode(int _ep){
    if(_ep <= 0) return;
    if(_ep > total_episode) return;

    playing_episode = _ep;
    playing_sec = 0;
}
```

หากมีการสั่งเปลี่ยนค่าผ่าน method นี้ไม่ถูกต้องค่าก็จะคงเดิม

```
a1.select_episode(-20);
a1.play(500);
a1.play(500);
```

Result

```
playing => [The Melancholy of Haruhi Suzumiya] Author => [Nagaru Tanigawa]
Episode => [EP.1] playing time =  [500 sec]
playing => [The Melancholy of Haruhi Suzumiya] Author => [Nagaru Tanigawa]
Episode => [EP.1] playing time =  [1000 sec]
```

ให้นักศึกษาแก้ไข โปรแกรม 2_3..CPP ให้ทำงานได้ถูกต้อง

ให้เวลา 15 นาที

ให้นักศึกษาลองเปลี่ยนค่า ตัวเลข ของคำสั่งนี้ ดังนี้

แล้วตรวจสอบผลลัพธ์ที่ได้

```
a1.set_playing_episode(-20);

a1.set_playing_episode(2);

a1.set_playing_episode(28);
```

สรุป Variable modifiers ทำให้เรากำหนดขอบเขตการเข้าถึงตัวแปรหรือฟังกชั่นได้เป็น

- Public เข้าถึงได้จากทุกที่
- Private เข้าถึงได้เพียงแค่ใน struct หรือ class เดียวกัน
- Protected เข้าถึงได้จาก class ที่สืบทอด (inheritance) ไป

ตัวอย่างกำหนดกลุ่มตัวแปรให้เป็นชนิดที่เราต้องการ

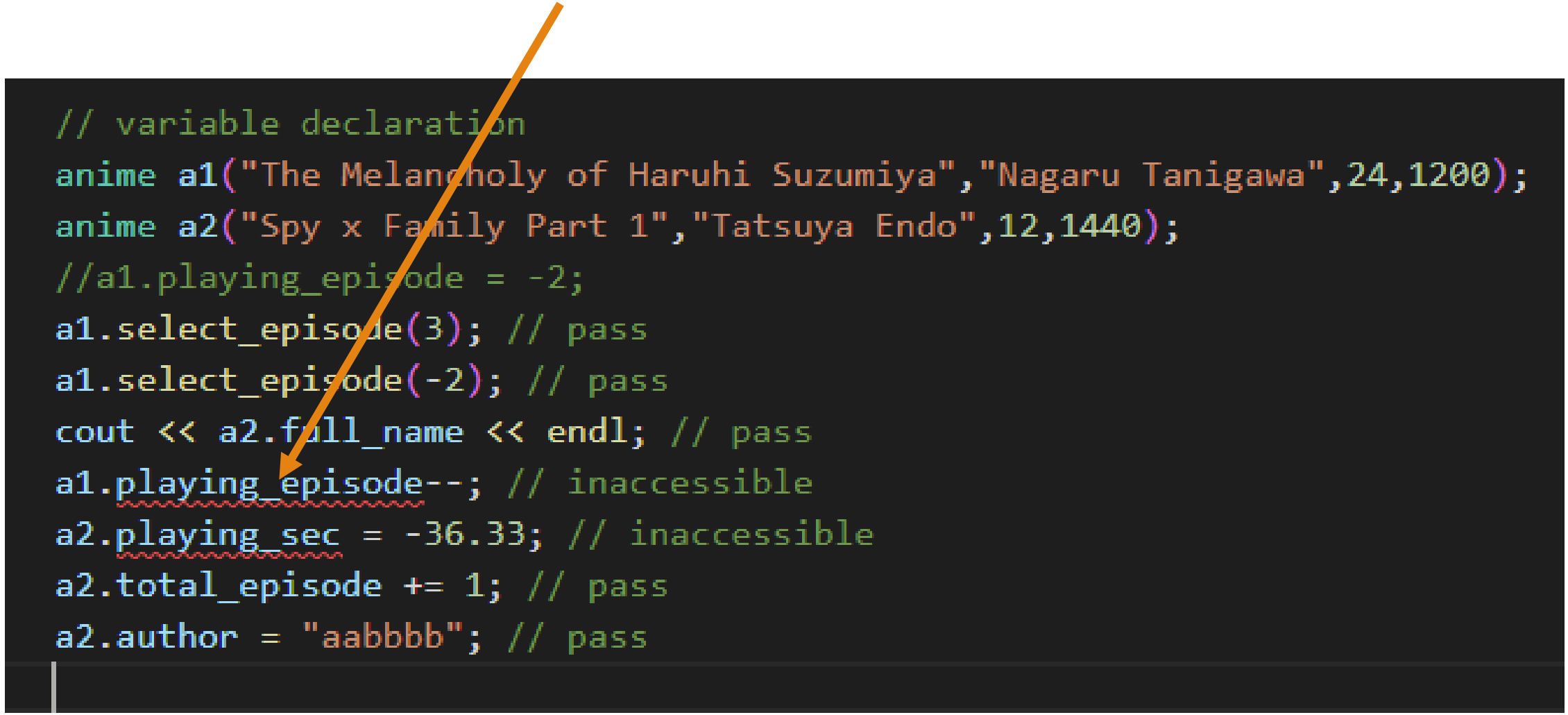
```
struct or class name{
public :
    public property1;
    public property2;
    public property3;
    public method1();
    public method2();

private :
    private property1;
    private property2;
    private property3;
    private method1();
    private method2();
}
```

Public section

Private section

หากกำหนดเป็น private แล้วก็จะเข้าถึงแบบปกติไม่ได้



```
// variable declaration
anime a1("The Melancholy of Haruhi Suzumiya","Nagaru Tanigawa",24,1200);
anime a2("Spy x Family Part 1","Tatsuya Endo",12,1440);
//a1.playing_episode = -2;
a1.select_episode(3); // pass
a1.select_episode(-2); // pass
cout << a2.full_name << endl; // pass
a1.playing_episode--; // inaccessible
a2.playing_sec = -36.33; // inaccessible
a2.total_episode += 1; // pass
a2.author = "aabbbb"; // pass
```

โปรแกรม 2_4...CPP struct anime จะมีการใส่ Deconstructor ด้วย

```
anime(string _name, string _author, int _ep, int length)
{
    cout << "this is constructor for " << _name << endl;

    full_name = _name;
    author = _author;
    total_episode = _ep;
    length_per_episode = length;

    playing_episode = 1;
    playing_sec = 0;
}

~anime()
{
    cout << full_name << " has destroyed" << endl;
}
```

Destructor คือ ฟังก์ชั่นหรือ method ที่ใช้ตอนปิดการใช้ Object โปรแกรม 2_4...CPP

- Create method that has ‘~’ symbol follow by class name
- สร้าง method ที่ชื่อขึ้นต้นด้วยเครื่องหมาย ‘~’ และตามด้วยชื่อ Class

```
~anime(){
    cout << full_name << "has destroyed" << endl;
}
```

Object จะถูกปิดเมื่ออยู่นอกขอบเขตที่ตนถูกสร้างขึ้น

```
anime a1("The Melancholy of Haruhi Suzumiya", "Nagaru Tanigawa", 24, 1200);

int main()
{
    system("cls");
    anime a1;
    anime a2("Spy x Family Part 1", "Tatsuya Endo", 12, 1440);
    if (true)
    {
        anime a3("detective conan", "Gosho Aoyama", 1067, 1200);
        a3.play(20);
    }
    a1.full_name = "aaa";
    a1.total_episode = -2;
    a1.playing_sec = -2;
    a1.select_episode(3);
    cout << "============" << endl;
    a1.play(500);
    a2.play(99999);
    cout << "++++++++++++" << endl;
    return 0;
}
```

a1 เป็น Object คนละตัว

a3 ถูกสร้างใช้งานในขอบเขตคำสั่ง
if { } เท่านั้นหากออกจากนี้ a3 ก็ถูกปิด

```
41      anime(){
42          cout << "this is default constructor" << endl;
43
44          playing_episode = 1;
45          playing_sec = 0;
46      }
47
48      anime(string _name,string _author,int _ep,int length){
49          cout << "this is constructor for " << _name << endl;
50
51          full_name = _name;
52          author = _author;
53          total_episode = _ep;
54          length_per_episode = length;
55
56          playing_episode = 1;
57          playing_sec = 0;
58      }
59
```

constructor

```
~anime(){
    cout << full_name << "has destroyed" << endl;
}
```

de-constructor

โปรแกรม 2_4.cpp สังเกตว่าจะ ปิดจากตัวที่สร้างหลังสุดใน main เป็นลำดับจนถึงตัวนอก main

```cpp
anime a1("The Melancholy of Haruhi Suzumiya", "Nagaru Tanigawa", 24, 1200);

int main()
{
    system("cls");
    anime a1;
    anime a2("Spy x Family Part 1", "Tatsuya Endo", 12, 1440);
    if (true)
    {
        anime a3("detective conan", "Gosho Aoyama", 1067, 1200);
        a3.play(20);
    }
    a1.full_name = "aaa";
    a1.total_episode = -2;
    a1.playing_sec = -2;
    a1.select_episode(3);
    cout << "============" << endl;
    a1.play(500);
    a2.play(99999);
    cout << "++++++++++++" << endl;
    return 0;
}
```

```
this is constructor for The Melancholy of Haruhi Suzumiya
this is default constructor
this is constructor for Spy x Family Part 1
this is constructor for detective conan
 playing => [detective conan] Author => [Gosho Aoyama]
 Episode => [EP.1] playing time =  [20 sec]
detective conan has destroyed
============
 playing => [AAA] Author => []
 Episode => [EP.1] playing time =  [498 sec]
 playing => [Spy x Family Part 1] Author => [Tatsuya Endo]
 Episode => [EP.1] playing time =  [1440 sec]
  ---> full Episode <---
++++++++++++
Spy x Family Part 1 has destroyed
AAA has destroyed
The Melancholy of Haruhi Suzumiya has destroyed
```

บทต่อๆไป ถ้าใช้ class แทน struct เราจะได้ค่าเริ่มต้นตัวแปรเป็น private

```
6   struct anime{
7   private :
8       int playing_episode; // last played episode
9       int playing_sec; // last played seconds in episode
10
11  public :
12      string full_name;
13      string author;
14      int total_episode;
15      int length_per_episode; // Average running time in Secon
16
17      void play(int time){ // play method
18          int remaining_time = length_per_episode - playing_se
19          if(time > remaining_time){ // next ep
20              cout << "playing " << full_name << " EP." << pla
```

```
6   class anime{
7       int playing_episode; // last played episode
8       int playing_sec; // last played seconds in episode
9
10  public :
11      string full_name;
12      string author;
13      int total_episode;
14      int length_per_episode; // Average running time in Second
15
16      void play(int time){ // play method
17          int remaining_time = length_per_episode - playing_sec
18          if(time > remaining_time){ // next ep
19              cout << "playing " << full_name << " EP." << play
```

Another Example โปรแกรม car...CPP

```
Class Car
Attribute :
    string name;
    float acceleration;
    float deceleration;
    float speed;
Method :
    void print()
    void speed_up()
    void speed_down()
```

```
car c1("ae86",1.5);
car c2;
car c3("Honda wave",0.2);
car c4("tesla model X",9.92);
```

```
cout << "speeding up" << endl;
for(int i=0;i<10;i++){
    c1.speed_up(); c2.speed_up(); c3.speed_up(); c4.speed_up();
}

c1.print(); c2.print(); c3.print(); c4.print();

cout << "slowing down" << endl;
c1.speed_down(); c2.speed_down(); c3.speed_down(); c4.speed_down();

c1.print(); c2.print(); c3.print(); c4.print();
```

โปรแกรม 2_5...CPP

LAB

- แก้ไขส่วน struct ให้โปรแกรมแสดงผลตามที่ comment สีเขียวไว้ในโปรแกรม
- ห้ามแก้ไขส่วนของคำสั่งใน main ( ) เด็ดขาด
- ก็อปปี้ไฟล์ที่แก้ไขแล้วส่งอาจารย์ใน msteams

(ให้ส่งทั้งโปรแกรม 2_3.CPP และ 2_5.CPP ใน msteams)