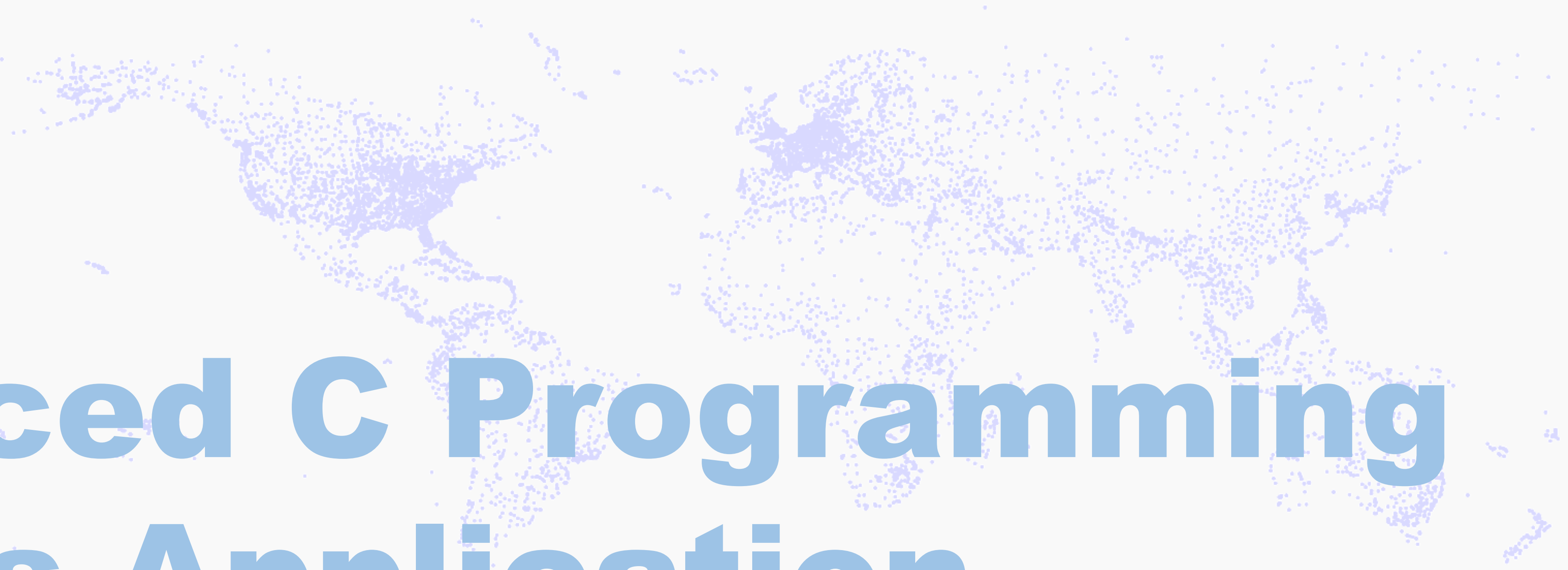


Advanced C Programming And It’s Application

Struct I

Assistant Prof. Chan, Chun-Hsiang

Department of Artificial Intelligence, Tamkang University

Dec. 29, 2021

<Outline/>

大綱

[1] Struct declaration

[2] typedef

[3] Create a struct

</Outline>

Why do we need a “struct” in C?

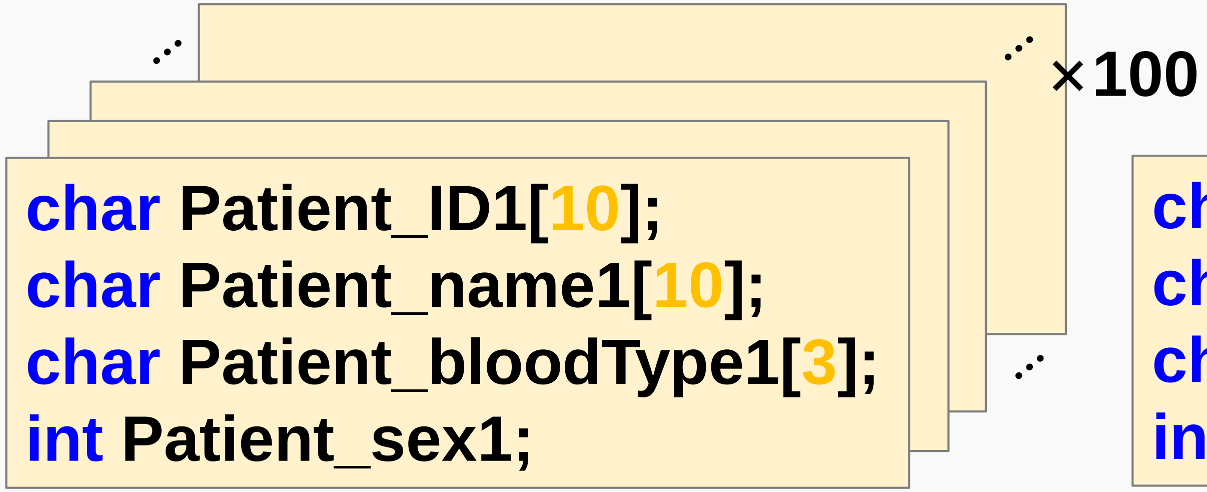
在C的世界中，我們可以利用「結構(struct) 」來儲存屬性資料；在C++的世界中，就可以直接用「類別(class) 」且有更強大的功能。那麼究竟甚麼是屬性資料呢?

例如:

一個病患進到醫院，如果是初診，我們都會要求填初次就診單，上面就有很多資訊要填，像是: 姓名、性別、血型、地址、電話等。這些資訊都是病患的屬性資料，之後給定一個獨立的病患識別碼(Patient ID)，作為方便尋找之用。

Why do we need a “struct” in C?

既然知道結構的重要性，其實我們也可以定義一連串的變數儲存這些資料，但是如果今天要做尋找病患資料的時候，你就需要去每個變數尋找他的欄位資料，變得很沒有效率。因此如果用結構的方式，就可以一起將所有資料調出來。



```
char Patient_ID1[10];
char Patient_name1[10];
char Patient_bloodType1[3];
int Patient_sex1;
```

Approach 1

```
char Patient_ID[100][10];
char Patient_name[100][10];
char Patient_bloodType[100][3];
int Patient_sex[100];
```

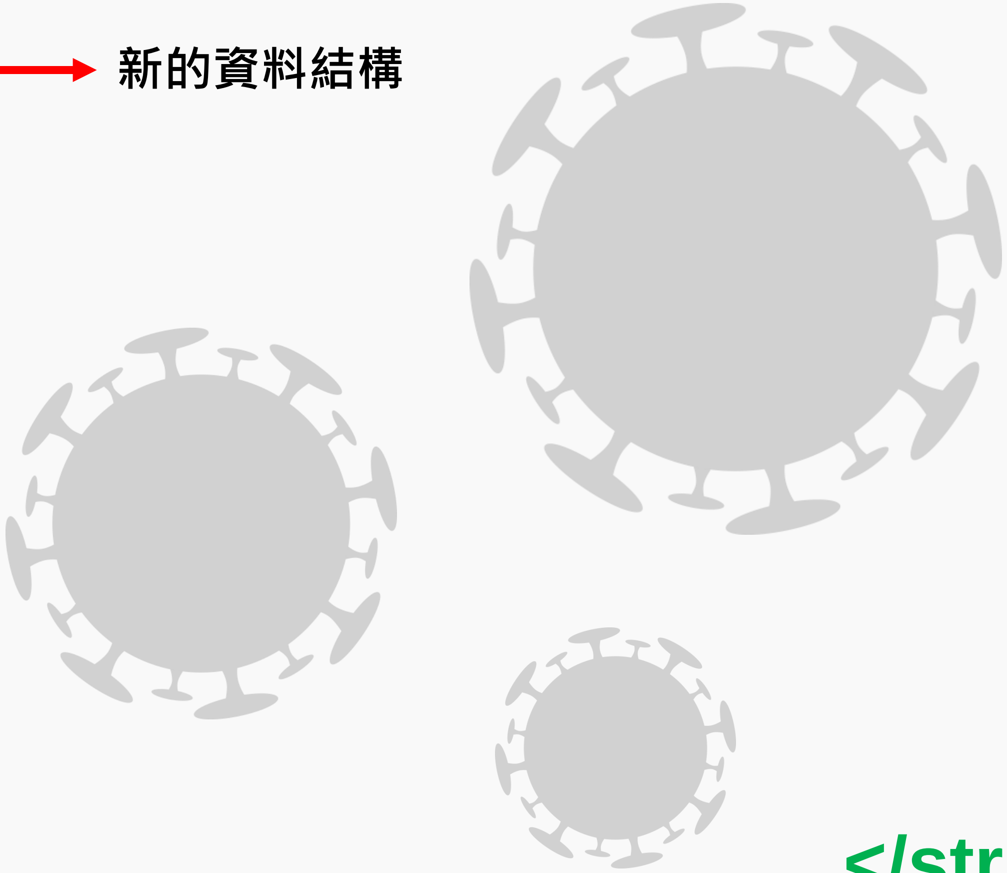
Approach 2

How to declare a “struct” in C?

但如果你是用結構來做的時候，就會變成特別簡單:

```
struct Patient {
    char Patient_ID[10];
    char Patient_name[10];
    char Patient_bloodType[3];
    int Patient_sex;
}
```

新的資料結構



How to declare a “struct” in C?

```
#include <stdio.h>
#include <string.h>
struct Flight{
        char flightNo[10];
        char airline[30];
        char origin[4], destination[4];
        int frequency, sitCapacity;
        double duration;
};
int main(){
        /*Ex 13-1: declare a struct in C */
        struct Flight EK367;
}
```

```
#include <stdio.h>
#include <string.h>
struct Flight{
        char flightNo[10];
        char airline[30];
        char origin[4], destination[4];
        int frequency, sitCapacity;
        double duration;
};
int main(){
        /*Ex 13-2: declare a struct in C++ */
        Flight EK367;
}
```

Using a struct is annoying?

你是否也覺得使用自定義的struct好像沒有想像中好用?

還要將struct再寫出來一次，的確有點annoying；所以這時候我們就有另一個工具 – typedef，他可以重新定義資料類別的顯示方式。

```
#include <stdio.h>
typedef int INT;
int main(){
        /*Ex 13-3: typedef */
        printf("/*Ex 13-3: typedef */\n");
        INT i = 10;
        printf("%d\n", i);
}
```

How to use typedef in struct?

```
#include <stdio.h>
#include <string.h>
typedef struct{
        char flightNo[10];
        char airline[30];
        char origin[4], destination[4];
        int frequency, sitCapacity;
        double duration;
} Flight;
int main(){
        /*Ex 13-4: typedef with struct method 1*/
        Flight EK367;
}
```

← 如果使用於struct會有這種效果。。。

How to use typedef in struct?

```
#include <stdio.h>
#include <string.h>
struct flight{
        char flightNo[10];
        char airline[30];
        char origin[4], destination[4];
        int frequency, sitCapacity;
        double duration;
};
typedef struct flight Flight;  🡸 也可以這樣做。。。
int main(){
        /*Ex 13-4: typedef with struct method 2*/
        Flight EK367;
}
```

How to use typedef in struct?

Lab 13-1:

想像你在監理所，需要有每一台有牌照的車輛資訊，所以此時你必須要宣告一個Car結構，需要含有以下的屬性資訊:

(1) char number[8]

(2) char driver[10]

(3) char brand[20]

(4) char carModel[20]

(5) int yearOfProduction

(6) int engineDisplacement

(7) double fuelConsumption

Create a struct

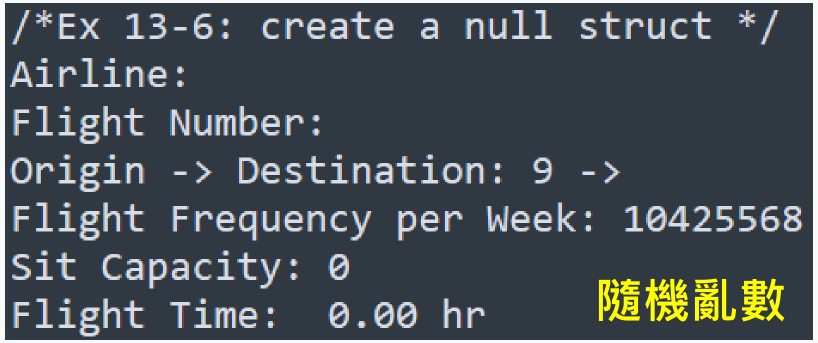
前面提到如果去建立一個struct的資料結構，再來就是要講如何將我們的資料放進，我們自定義的結構中。如果我們需要獲取該屬性資料的數值時，可以利用dot(.)來做index，再依照其資料型態，選擇適合的方式填入，像是整數就可以直接用等於；但如果是字串就需要利用strcpy的方式填入資料。

(1) 如果你甚麼都沒放，預設初始值就是亂數。
(2) 複製struct，可直接用等於將所有屬性複製到另一個struct中
(3) 輸入值也可以利用inline的方式，一口氣將所有資料填入

Create a null struct

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int main(){
        /*Ex 13-6: create a null struct */
        printf("/*Ex 13-6: create a null struct */\n");
        Flight EK367;

        printf("Airline: %s\nFlight Number: %s\n", EK367.flightNo, EK367.airline);
        printf("Origin -> Destination: %s -> %s\n", EK367.origin, EK367.destination);
        printf("Flight Frequency per Week: %d\n", EK367.frequency);
        printf("Sit Capacity: %d\n", EK367.sitCapacity);
        printf("Flight Time: %5.2lf hr\n", EK367.duration);}
```



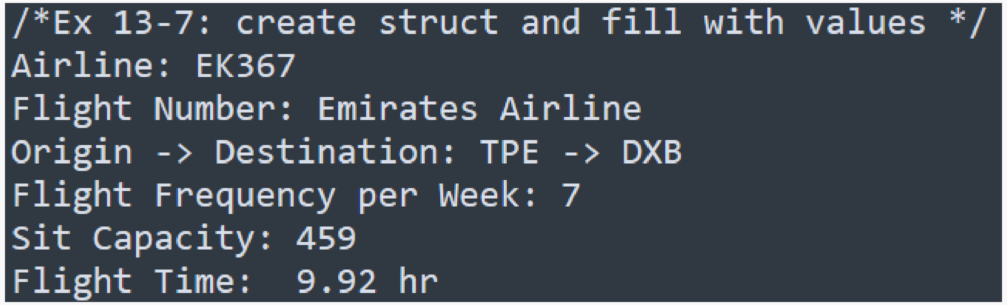
```
/*Ex 13-6: create a null struct */
Airline:
Flight Number:
Origin -> Destination: 9 ->
Flight Frequency per Week: 10425568
Sit Capacity: 0
Flight Time:  0.00 hr
```

隨機亂數

使用dot的方式便可獲得該屬性的資料

</create>

Create a struct and fill with values



```
/*Ex 13-7: create struct and fill with values */
Airline: EK367
Flight Number: Emirates Airline
Origin -> Destination: TPE -> DXB
Flight Frequency per Week: 7
Sit Capacity: 459
Flight Time:  9.92 hr
```

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int main(){
        /*Ex 13-7: create struct and fill with values */
        printf("/*Ex 13-7: create struct and fill with values */\n");
        Flight EK367;
        strcpy(EK367.flightNo, "EK367");
        strcpy(EK367.airline, "Emirates Airline");
        strcpy(EK367.origin, "TPE");
        strcpy(EK367.destination, "DXB");
        EK367.frequency = 7; EK367.sitCapacity = 459; EK367.duration = 9.917;
        printf("Airline: %s\nFlight Number: %s\n", EK367.flightNo, EK367.airline);
        printf("Origin -> Destination: %s -> %s\n", EK367.origin, EK367.destination);
        printf("Flight Frequency per Week: %d\n", EK367.frequency);
        printf("Sit Capacity: %d\n", EK367.sitCapacity);
        printf("Flight Time: %5.2lf hr\n", EK367.duration);}
```

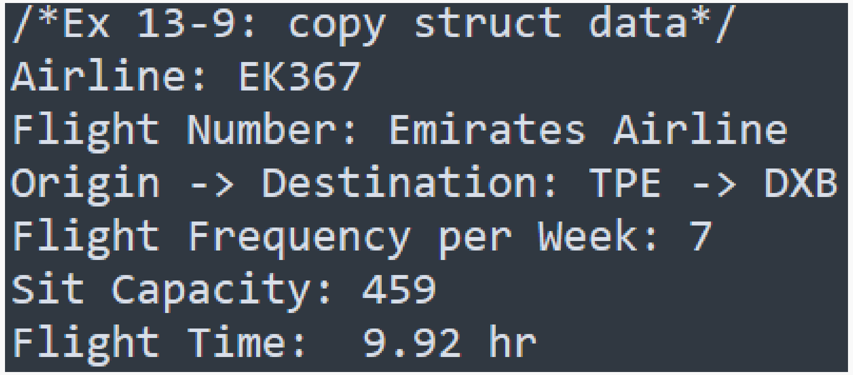
使用dot的方式便可獲得該屬性的資料

Create two struct data

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int main(){
        /*Ex 13-8: create two struct data*/
        printf("/*Ex 13-8: create two struct data */\n");
        Flight EK367, EK366;
        strcpy(EK367.flightNo, "EK367");
        strcpy(EK367.airline, "Emirates Airline");
        strcpy(EK367.origin, "TPE"); strcpy(EK367.destination, "DXB");
        EK367.frequency = 7; EK367.sitCapacity = 459; EK367.duration = 9.917;
        strcpy(EK366.flightNo, "EK366");
        strcpy(EK366.airline, "Emirates Airline");
        strcpy(EK366.origin, "DXB"); strcpy(EK366.destination, "TPE");
        EK366.frequency = 7; EK366.sitCapacity = 459; EK366.duration = 7.917;
}
```

使用dot的方式便可獲得該屬性的資料

Copy struct data

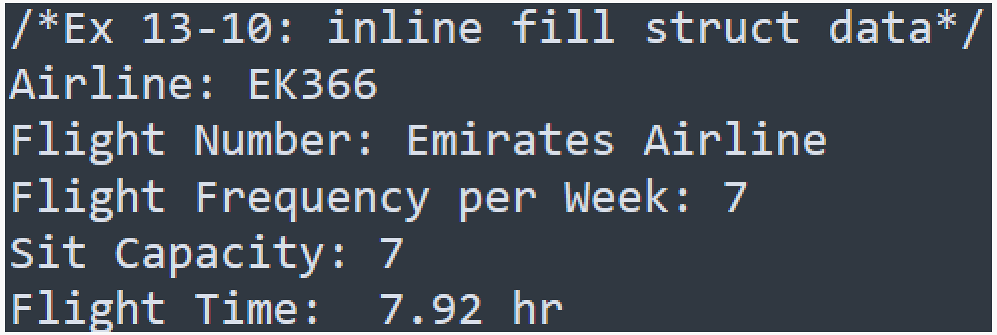


```
/*Ex 13-9: copy struct data*/
Airline: EK367
Flight Number: Emirates Airline
Origin -> Destination: TPE -> DXB
Flight Frequency per Week: 7
Sit Capacity: 459
Flight Time:  9.92 hr
```

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int main(){
        /*Ex 13-9: copy struct data*/
        printf("/*Ex 13-9: copy two struct data */\n");
        Flight EK367;
        strcpy(EK367.flightNo, "EK367");strcpy(EK367.airline, "Emirates Airline");
        strcpy(EK367.origin, "TPE"); strcpy(EK367.destination, "DXB");
        EK367.frequency = 7; EK367.sitCapacity = 459; EK367.duration = 9.917;
        Flight EK364 = EK367;
        printf("Airline: %s\nFlight Number: %s\n", EK364.flightNo, EK364.airline);
        printf("Origin -> Destination: %s -> %s\n", EK364.origin, EK364.destination);
        printf("Flight Frequency per Week: %d\n", EK364.frequency);
        printf("Sit Capacity: %d\n", EK364.sitCapacity);
        printf("Flight Time: %5.2lf hr\n", EK364.duration);}
```

Inline fill struct data

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int main(){
        /*Ex 13-10: inline fill struct data*/
        printf("/*Ex 13-10: inline fill struct data*/\n");
        Flight EK367;
        Flight EK367 = {"EK367", "Emirates Airline", "TPE", "DXB", 7, 459, 9.917},
               EK366 = {"EK366", "", "DXB", "TPE", 0, 0, 7.917};
        strcpy(EK366.airline, EK367.airline);
        EK366.frequency = EK367.frequency;
        EK366.sitCapacity = EK367.sitCapacity;
        printf("Airline: %s\nFlight Number: %s\n", EK366.flightNo, EK366.airline);
        printf("Flight Frequency per Week: %d\n", EK366.frequency);
        printf("Sit Capacity: %d\n", EK366.sitCapacity);
        printf("Flight Time: %5.2lf hr\n", EK366.duration);}
```



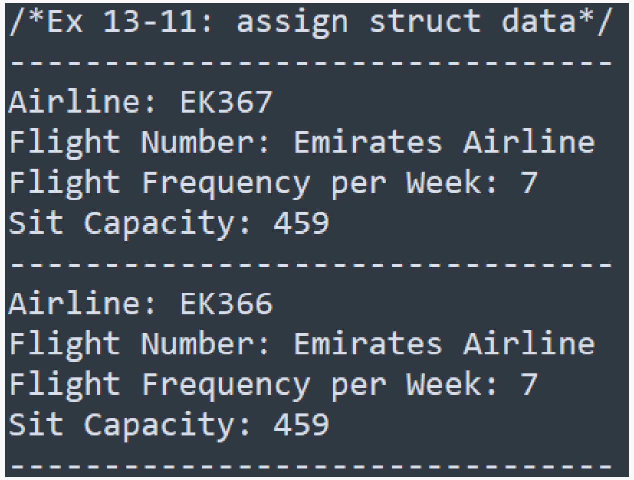
```
/*Ex 13-10: inline fill struct data*/
Airline: EK366
Flight Number: Emirates Airline
Flight Frequency per Week: 7
Sit Capacity: 7
Flight Time:  7.92 hr
```

<create/>

Assign struct data

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int main(){
        /*Ex 13-11: inline fill struct data*/
        printf("/*Ex 13-11: inline fill struct data*/\n");
        Flight EK367;
        Flight EK367 = {"EK367", "Emirates Airline", "TPE", "DXB", 7, 459, 9.917},
               EK366 = {"EK366", "Emirates Airline", "DXB", "TPE", 7, 459, 7.917};

        printf("Airline: %s\nFlight Number: %s\n", EK367.flightNo, EK367.airline);
        printf("Flight Frequency per Week: %d\n", EK367.frequency);
        printf("Sit Capacity: %d\n", EK367.sitCapacity);
        printf("Airline: %s\nFlight Number: %s\n", EK366.flightNo, EK366.airline);
        printf("Flight Frequency per Week: %d\n", EK366.frequency);
        printf("Sit Capacity: %d\n", EK366.sitCapacity);
```



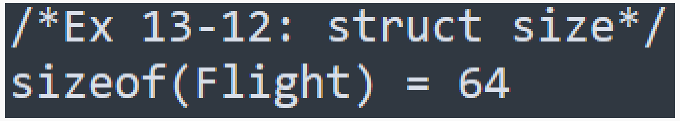
```
/*Ex 13-11: assign struct data*/
--------------------------------
Airline: EK367
Flight Number: Emirates Airline
Flight Frequency per Week: 7
Sit Capacity: 459
--------------------------------
Airline: EK366
Flight Number: Emirates Airline
Flight Frequency per Week: 7
Sit Capacity: 459
--------------------------------
```

</create>

struct size measurement

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int main(){
        /*Ex 13-12: struct size*/
        printf("/*Ex 13-12: struct size*/\n");
        Flight EK367;
        Flight EK367 = {"EK367", "Emirates Airline", "TPE", "DXB", 7, 459, 9.917};
        printf("sizeof(Flight) = %lu\n", sizeof(EK367));
```

Lab 13-2:
這邊顯示的大小為 64 bytes，請問是如何計算出來?



```
/*Ex 13-12: struct size*/
sizeof(Flight) = 64
```

Assign struct data

Lab 13-3:

將以下資訊以Lab13-1的Car struct做宣告，變數名稱可以使用entry欄位的資訊。

Entry	number	driver	brand	carModel	yearOfProduction	engineDisplacement	fuelConsumption
Car01	ABC-8888	Jon	Toyota	Sienta5	2020	1600	18.2
Car02	DEF-1314	Anna	Benz	E320	2019	3200	19.2
Car03	BBB-8888	Amy	BMW	X5	2018	3200	15.2
Car04	168-ABC6	Bon	Volvo	XC-40	2012	2400	8.5
Car05	404-NO56	Duke	BMW	X4	2012	2400	7.6
Car06	501-FOR8	Ray	Benz	C200	2016	2000	12.3

<References/>

參考資料

1. https://zh.wikipedia.org/wiki/%E7%BB%93%E6%9E%84%E4%BD%93_(C%E8%AF%AD%E8%A8%80)
2. https://openhome.cc/Gossip/CGossip/StructABC.html
3. https://ithelp.ithome.com.tw/questions/10190575
4. https://blog.xuite.net/tzeng015/twblog/113271886
5. 蔣宗哲教授講義

</References>