

Week 14

Structure

What's structure

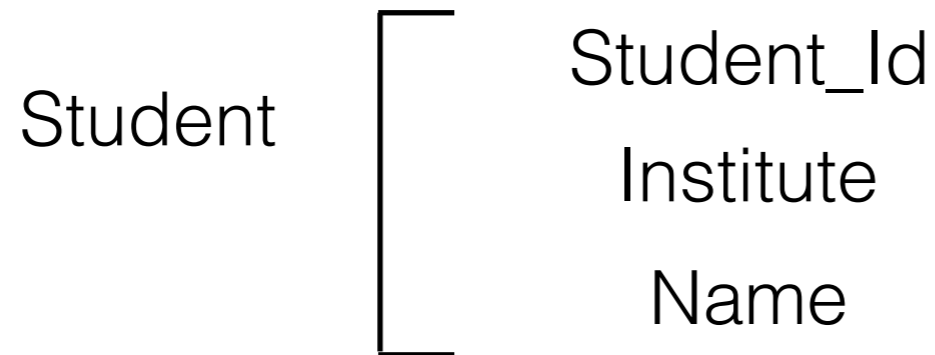
- A new data type

int, double, char, etc

structure



Why structure



If we want to store 5 student's information

`int Student_Id[5]`

`char Institute[5][20]` →

`char Name[5][20]`

`Student students[5]`



`int Student_Id`

`char Institute[20]`

`char Name[20]`

How to define

```
temp variable
struct t_point {
    int x;
    int y; data
};
typedef struct t_point Point;
```

上面的寫法
也可以合併成

```
typedef struct {
    int x;
    int y;
} Point;
```

define struct t_point type
into Point type

```
struct t_student {
    int Student_Id;
    char Institute[20];
    char Name[20];
};
typedef struct t_student Student;
```

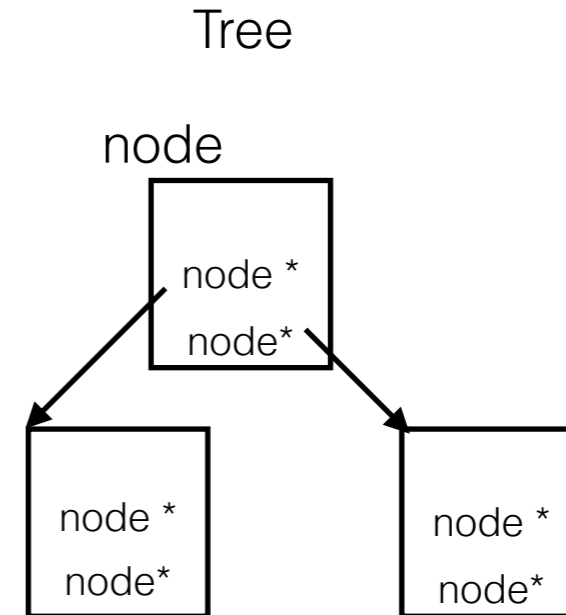
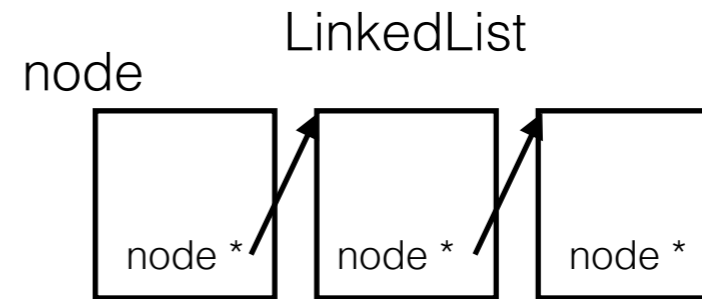
```
typedef int MyInt;
```

```
typedef struct {
    int Student_Id;
    char Institute[20];
    char Name[20];
} Student;
```

How to define

```
struct t_point {  
    struct t_point *ppap;  
    int x;  
    int y;  
};  
typedef struct t_point Point;
```

```
typedef struct t_point {  
    struct t_point *ppap;  
    int x;  
    int y;  
} Point;
```



How to use

可看成一維陣列



```
Point pt = {5, 7}; 直接賦值  
Point *pp;
```

```
pp = (Point *)malloc(sizeof(Point)*10)
```

```
pp = &pt;
```

對pp這個pointer取值

```
(*pp).x = 10;
```

然後再拿裡面的x

```
pp->x = 10;
```

直接從pp這個pointer拿裡面的x

more easy to use



Example 1

```
#include <stdio.h>
#include <stdlib.h>
/* 定義一個新的型別 */
/* 取名叫做 Point */
/* 裡面包含 x 和 y 兩個 members */
/* 定義過之後 Point 可以被拿來當作一般的型別來使用 */
/* 包括宣告新的變數 或是宣告 function */
typedef struct {
    int x;
    int y;
} Point;

/* ones_vec_1() 會傳回某個 Point 結構的位址 */
/* 這個位址是由 Point 結構組成的陣列的開頭位址 */
Point * ones_vec_1(int length);

void ones_vec_2(int length, Point **bp);
```

開一個長度為length的Point
陣列，並回傳其structure陣
列開頭位址

傳入一個開頭位址，並在此開頭位址
建立一個Point陣列

Example 1

```
int main(void)
{
    Point *a, *b;
    int i, length;

    printf("The size of a Point is %u bytes.\n", sizeof(Point));

    printf("vector length: ");
    scanf("%d", &length);

    /* 利用 ones_vec_1 取得一個陣列 */
    /* 陣列的每個元素是一個 Point */
    /* 陣列的開頭位址記錄在指標變數 a 裡面 */
    a = ones_vec_1(length);
    ones_vec_2(length, &b);

    /* a 是個指標變數 它記錄的是某個陣列的開頭位址 */
    /* 陣列的每個元素是 a[i] (是個 Point) */
    /* 所以有兩個 members 分別是 a[i].x 和 a[i].y */
    for (i=0; i<length; i++)
        printf("(%d, %d) ", a[i].x, a[i].y);
    printf("\n");

    for (i=0; i<length; i++)
        printf("(%d, %d) ", b[i].x, b[i].y);
    printf("\n");

    return 0;
}
```

int x -> 4 bytes
int y -> 4 bytes
total 8 bytes

傳入b這個Point的開頭位址，用&b取位址

Example 1

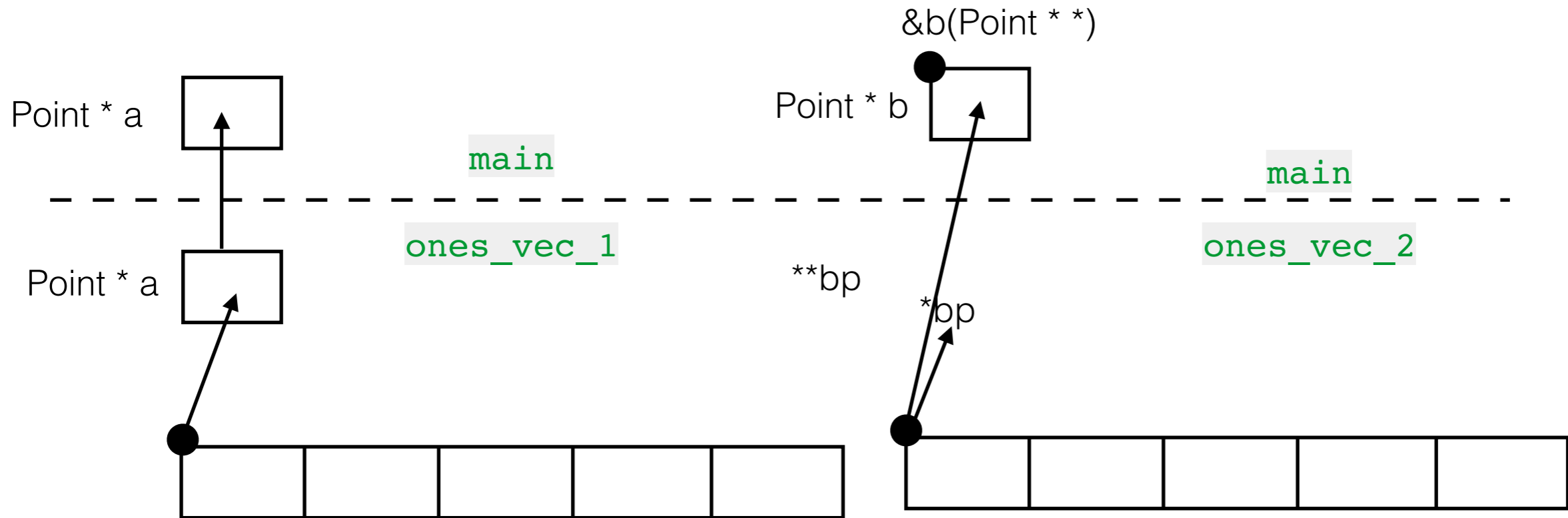
```
Point * ones_vec_1(int length)
{
    Point *a;
    int i;
    a = (Point *) malloc(length * sizeof(Point));
    for (i = 0 ; i < length; i++) {
        a[i].x = 1;
        a[i].y = 1;
    }
    return a;
}
```

傳入的是Point*這個指標個開頭
位址，所以是Point**

```
void ones_vec_2(int length, Point **bp)
{
    int i;

    *bp = (Point *) malloc(length * sizeof(Point));
    for (i = 0 ; i < length; i++) {
        (*bp)[i].x = 1;
        (*bp)[i].y = 1;
    }
}
```

How it works



Example2

```
#include <stdio.h>
#include <stdlib.h>
typedef struct t_complex {
    double r;
    double i;
} Complex;

void add(Complex a, Complex b, Complex *t)
{
    t->r = a.r+b.r;
    t->i = a.i+b.i;
}
void set_complex(Complex *p, double r, double i)
{
    p->r = r;
    p->i = i;
}
void show_complex(Complex t)
{
    printf("%.2f%+.2fi\n", t.r, t.i);
}

int main(void)
{
    Complex x, y, z;
    set_complex(&x, 1, 2);
    set_complex(&y, 2, -3);
    add(x, y, &z);
    show_complex(z);

    return 0;
}
```

傳入Complex a

a.r / a.i取值

傳入Complex *a

(*a).r / (*a).i 或 a->r / a->i
取值