slide แรก

st

char

int

};

ดูใน slide หาก อ. ถูก

int main() {   size of ของ simple1 คือ 8

struct example simple1 ;

struct card aCard1 = aCard2   เป็นการ copy ค่า

slide 2   เพิ่มด้วน

struct fraction {

int numerator;

int denominator;   ใส่ int de... = 1; จะ error

};

13 n.w. 63

ex

ex 예제

```
typedef struct {
    int width;
    int length;
    int height;
} Box;
typedef struct {
    int radius;
} Circle;
int main() {
    Box mybox;
    Circle c;
```

```
mybox.width = 10;
my.box.length = 30;
mybox.height = 10;
c.radius = 10;
printf("Box (%d, %d, %d)\n", mybox.width,
    mybox.length, mybox.height)
printf("Circle (%d)\n", c.radius)
```

⇒ Box (10, 30, 10)

⇒ Circle (10)

slide 2 เติมส่วน

```
struct fraction {
    int numerator;
    int denominator;    ใส่ int de... = 1; จะ error
};

int main() {
    struct fraction half = {1, 2};
    s      n        eight = {8};
⇒ 1/2    printf("%d / %d\n", half.num~, half.de...);
⇒ 8/0        "             eight.num~, eight.de...);
    struct fraction another = half;
1/2      printf("%d/%d \n", half.num~, half.de~);   (half)
1/2      print(another ~, anoter.num, anothe.de~);
ก่อนค่า   { half.num~ = 2;
จากข้างบน  { half.de~ = 4;
จะเปลี่ยน
↳ 2/4    printf(half ~, half.num~, half.de~);
1/2      printf(another ~, another.num, another.de~);
```

sprintf (p → name , "part %d", i);

└ เอาค่าที่ print ไปใส่ในนี้ └ เหมือน printf ธรรมดา

คือ เอา part %d ไปใส่ใน name แต่ละตัวแปล

```
{                          ~ {
struct part a,b;           →
b.price = 39.99;           →
b.am = "floppy";           →
a = b;                     memcpy (&a, &b, sizeof(part));
}                          }
```

ความหมาย เหมือนกันคือ copy ค่า b ไปใส่ใน a ด้วย

ประกาศไว้ก่อน ในกรณี struct B อยู่ใน A ได้

```
struct B;
struct A {
    struct B * partner;
};
struct B {
    struct A * partner;
};
```

เป็น pointer

ถ้าไม่เป็น pointer ก็ได้

อ่านสไลด์ 2 เพิ่ม เติม (ดูเป็นเทพ เพื่อความเข้าใจง่าย)

13 ก.พ. 63

11 กพ 63

๑. ตัวอย่าง การประกาศ Structure

```
struct employee2 {
    char _ [20];
    char _ [20];
    int _;
};
struct employee2 person   // ERROR
       *a_ptr             // Pointer

struct example {
    char c;
    int i;
} simple1, simple2;
```

```
    return b.wid~ * b.h~ * b.le~ ;
}
int main()
{
    struct Box b;
    printf(" Enter the Box ~(~ ): ");
    scanf("%d %           &b.wi~ , &b.h~ , &b.le~ );
```

การเปรียบเทียบ อยู่ในสไลด์ ไปอ่าน

Array ~ ในสไลด์ เหมือนกัน อ่านเอา

Size of a Struct

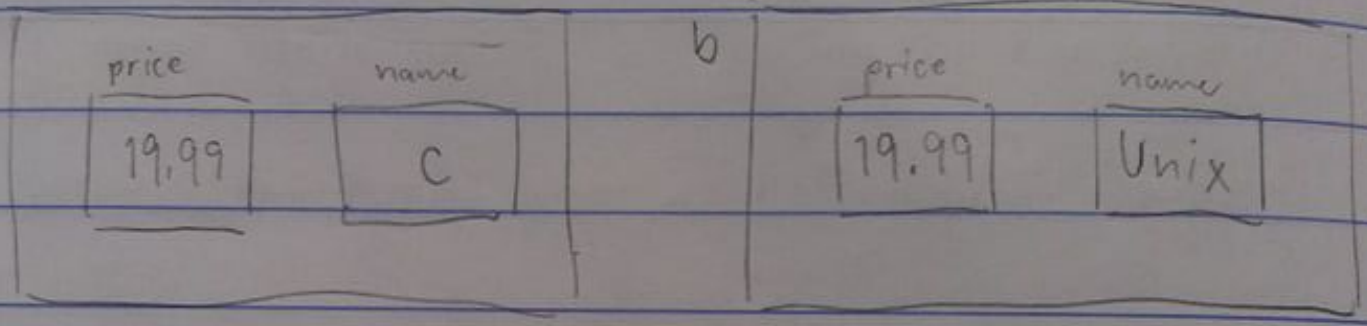
* sizeof() ออกสอบ ในสไลด์ ลองอ่าน ท่อง บวกกันได้ 26

⇒ strcpy (b.name, "Unix book");

puts (a.name);

puts (b.name);

}



⇒ a = b ก็อ สร้าง ▭ a ขึ้นมาข้างในเหมือน b เลย

⇒ ก็ตงค่า b.name อันใหม่ลงไป

อันนี้ไม่ใช้ pointer ค่าเลยเปลี่ยน ถ้าใช้ pointer ค่าจะออกมาเป็นเหมือนกัน (Unix เหมือนกัน) เพราะ b และ a ชี้ไปที่ ค่าๆเดียวกัน ถ้า ค่านั้น โดนเปลี่ยน

20 ก.พ. 2563

ex Enter the box dimensions (width, length, height) : 1 2 3

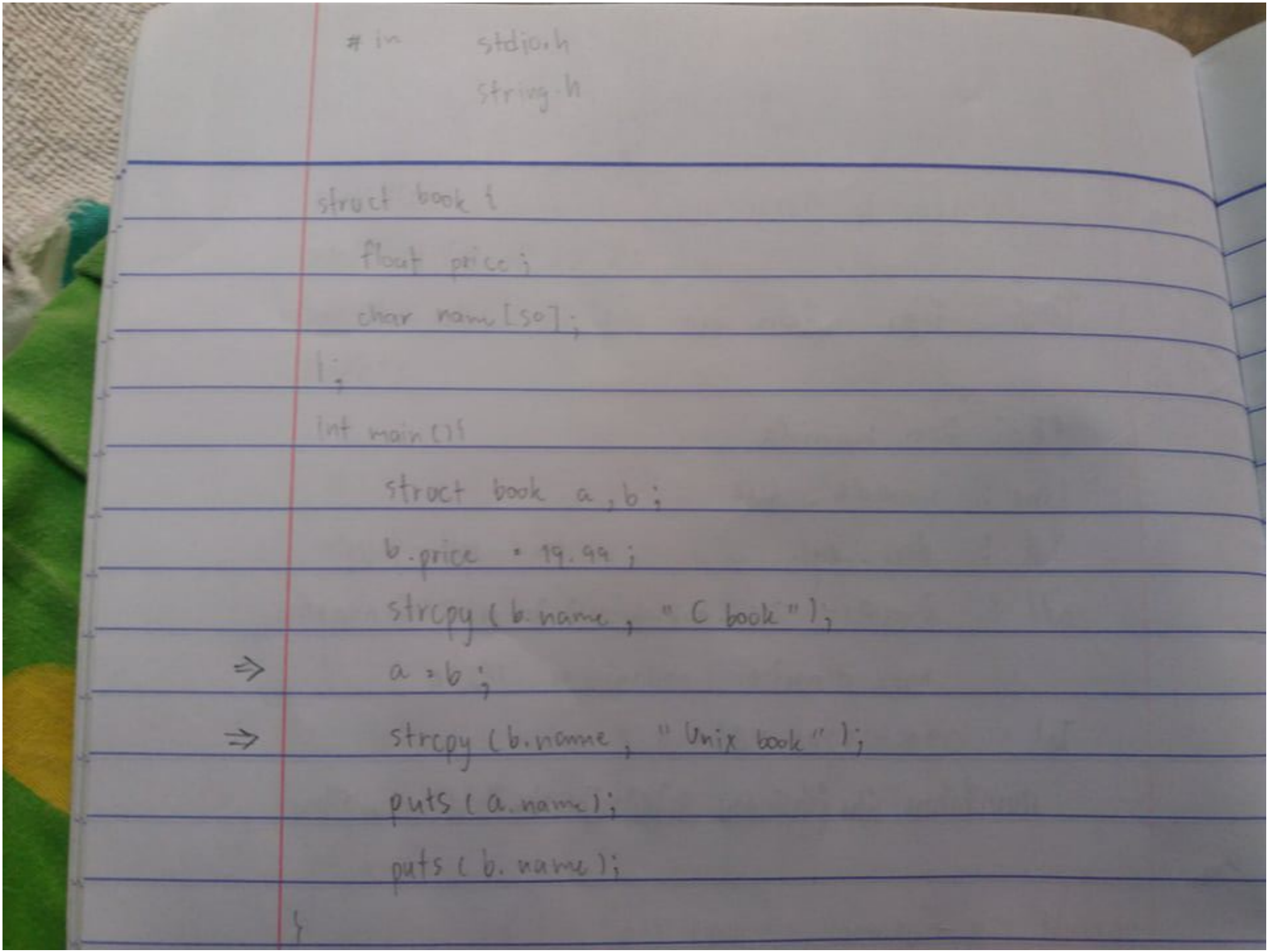
Enter the radius of the circle : 0.8

Box volume : 6

Circle area : 2.01

```
typedef struct {
    int wid_, le_, h_;
} Box;
typedef struct {
    int radius;
} Circle;
int main() {
    Box mybox;
    Circle c;
```

abstraction การกำหนดลักษณะสำคัญ



```
#in     stdio.h
        string.h

struct book {
   float price;
   char name[50];
};
int main(){
    struct book a, b;
    b.price = 19.99;
    strcpy(b.name, "C book");
⇒   a = b;
⇒   strcpy(b.name, "Unix book");
    puts(a.name);
    puts(b.name);
}
```

```
Printf ("Enter ____ : ")
scanf ("%d %d %d", &w, &l, &h)
printf("Enter radius ___ : ")
scanf("%
#
#
struct Box { int w~, h~, l~ };
int Get Volume ( struct Box b)
{
    return b.wid~ * b.h~ * b.le~ ;
}
int main()
{
    struct Box b;
    printf (" Enter the Box ~( ~ ) : ");
```