Structures (1A)

Young Won Lim
12/4/17

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice.

Young Won Lim
12/4/17

Structure Variable Declaration

structure type

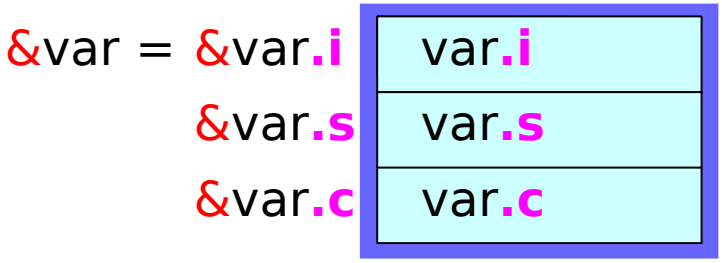
```
struct aaa {
        int     i;
        short   s;
        char    c;
} ;
```

definition

```
struct aaa  var;
```

var declaration

accessing a structure variable



accessing its members

Compare with function definitions

structure type

```
struct aaa {
        int         i;
        short       s;
        char        c;
} ;
```

definition

```
struct aaa    var;
```

var declaration

```
var.i = 100;
```

member access

function type

```
int add (int x, int y) ;
```

prototype

```
int add (int x, int y) {
  return (x + y);
}
```

definition

```
sum = add (30, 20) ;
```

function call

Struct Variable Declaration (1)

structure type

```
struct aaa {
        int     i;
        short   s;
        char    c;
} ;
```

definition

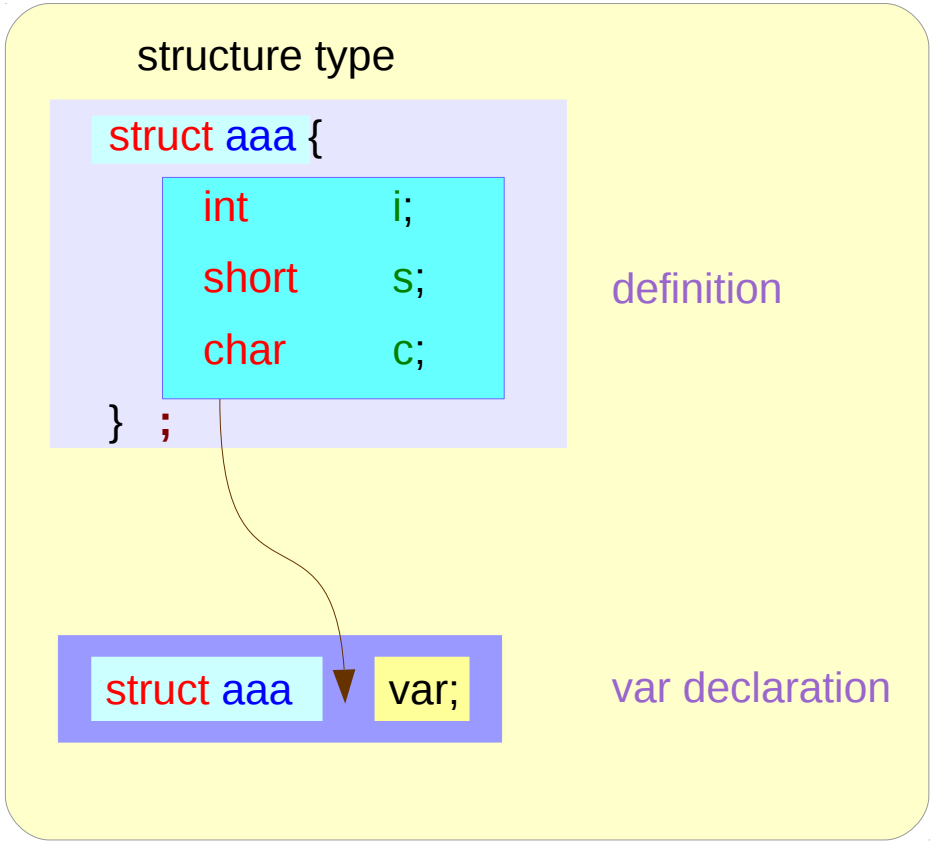
```
struct aaa  var;
```

var declaration

structure type

```
struct aaa {
        int     i;
        short   s;
        char    c;
} var ;
```

Struct Variable Declaration (2)




```
struct aaa {
        int     i;
        short   s;
        char    c;
} var ;
```

structure type definition

+ variable declaration

Struct Variable Declaration (3)

structure type

```
struct aaa {
        int         i;
        short       s;
        char        c;
} ;
```

```
typedef struct aaa   ATYPE ;
```

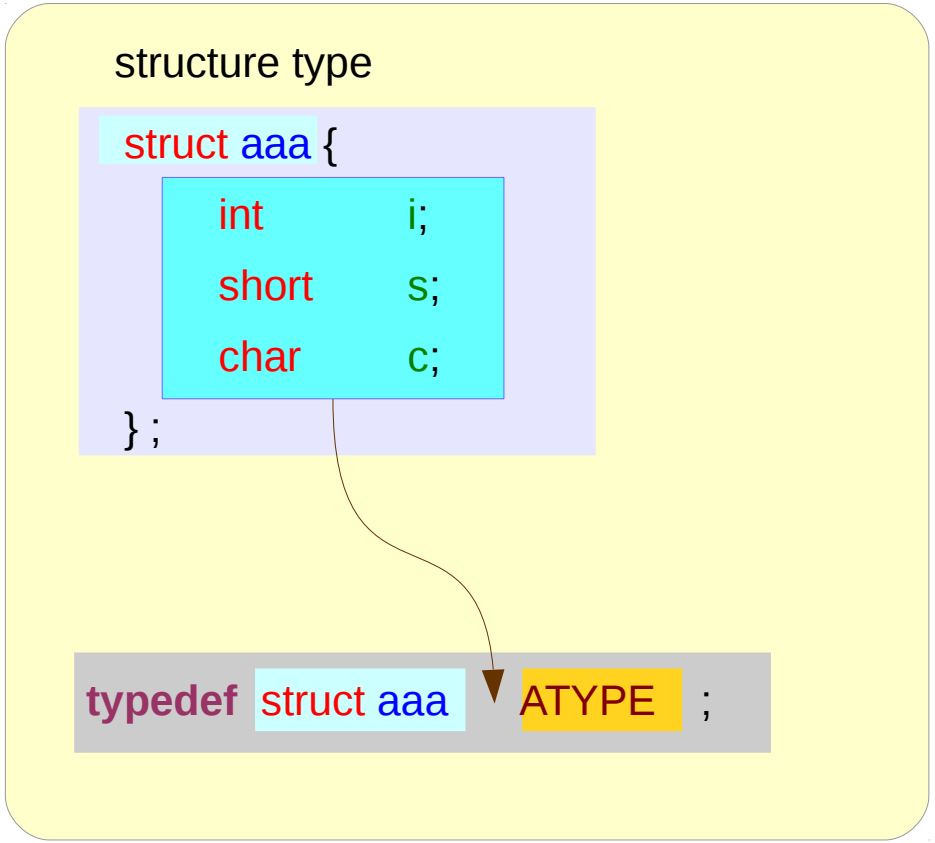
```
ATYPE       var;
```

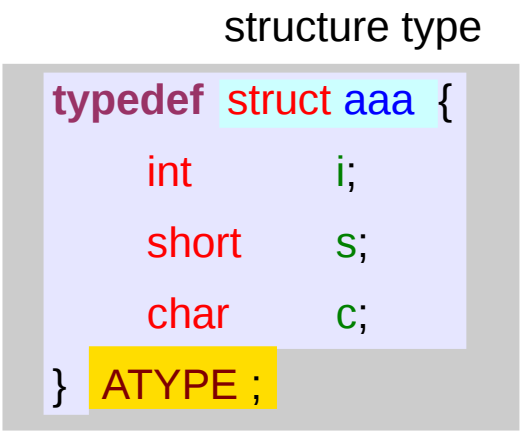
structure type

```
typedef struct aaa {
        int         i;
        short       s;
        char        c;
} ATYPE ;
```

```
ATYPE       var;
```

Struct Variable Declaration (4)






```
ATYPE var;
```

```
ATYPE var;
```

Struct Variable Declaration Summary

structure type

```
struct aaa {
        int     i;
        short   s;
        char    c;
} ;
```

```
struct aaa  var;
```

structure type

```
struct aaa {
        int     i;
        short   s;
        char    c;
} ;
```

```
typedef struct aaa  ATYPE ;
```

```
ATYPE  var;
```

structure type

```
struct aaa {
        int     i;
        short   s;
        char    c;
} var ;
```

structure type

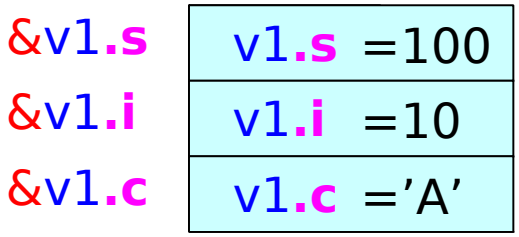
```
typedef struct aaa {
        int     i;
        short   s;
        char    c;
} ATYPE ;
```

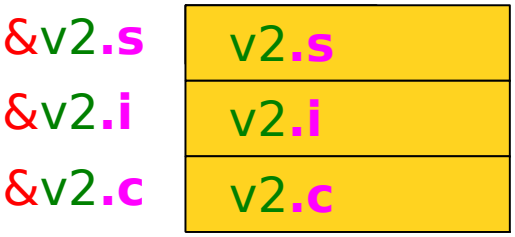
```
ATYPE  var;
```

Initialization

structure type

```
struct aaa {
        short       s;
        int         i;
        char        c;
} ;
```

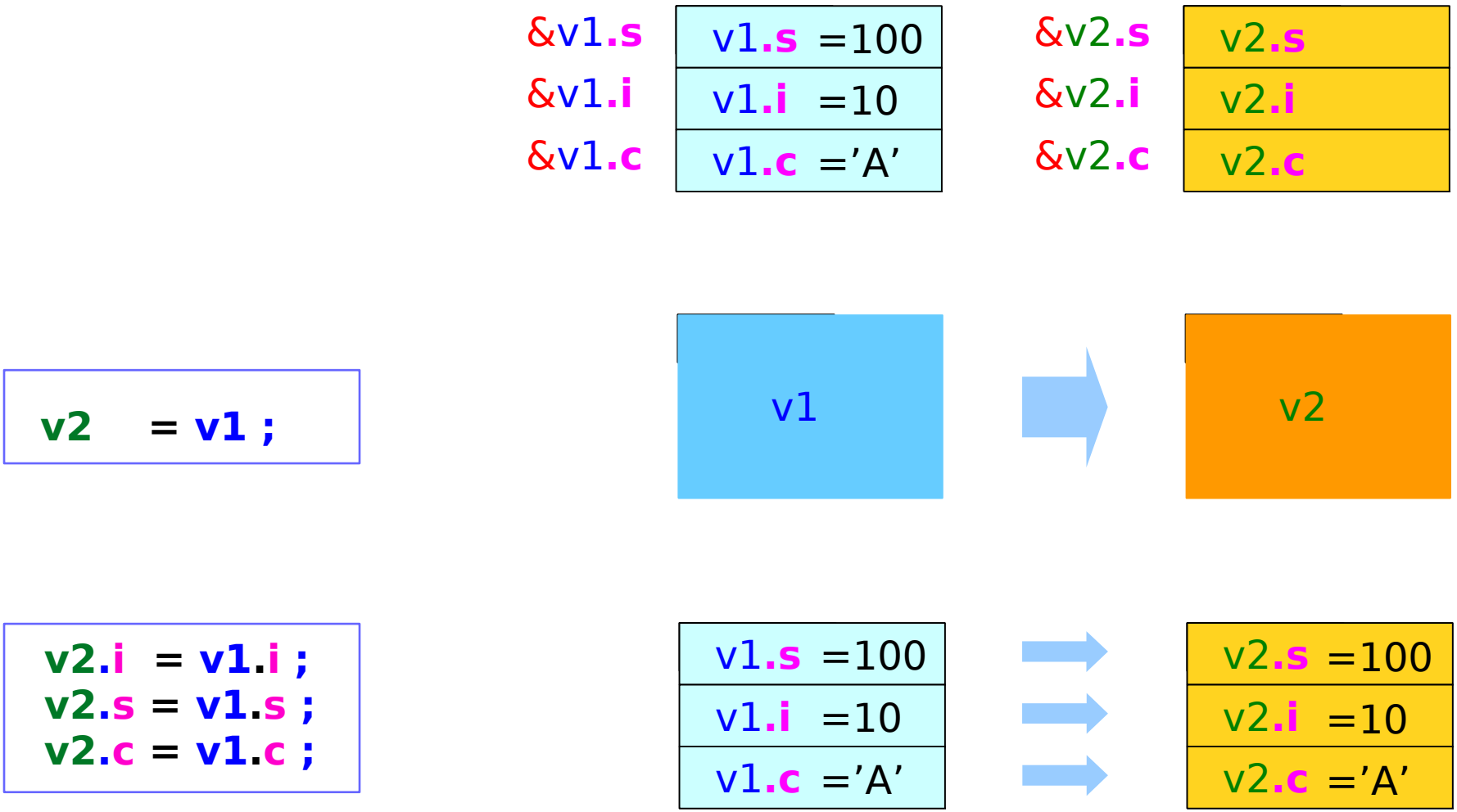





```
struct aaa    v1 = { 100, 10, 'A'} ;
```

```
struct aaa    v2;
```

Assignments



Variable Declaration Allocates Memory

structure type

```
struct aaa {
        short       s;
        int         i;
        char        c;
} ;
```

definition

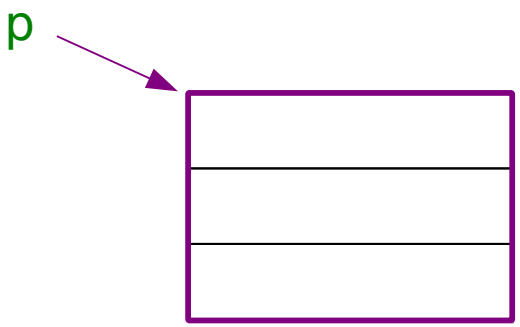
```
struct aaa    v1 = { 100, 10, 'A'} ;
```

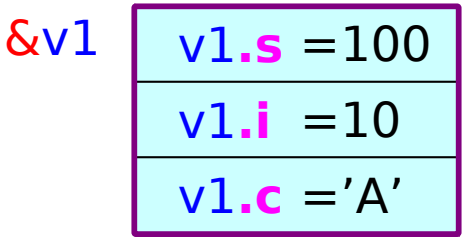
&v1

v1.s =100
v1.i =10
v1.c =’A’

Pointer to Structure Variable Declaration

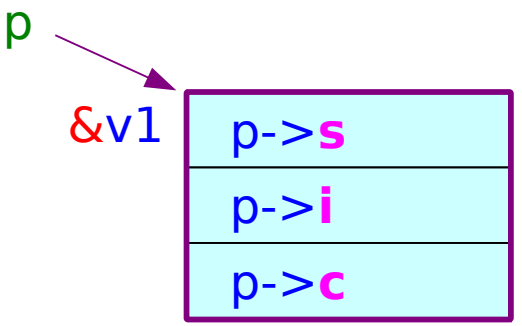
```
struct aaa   *p ;
```





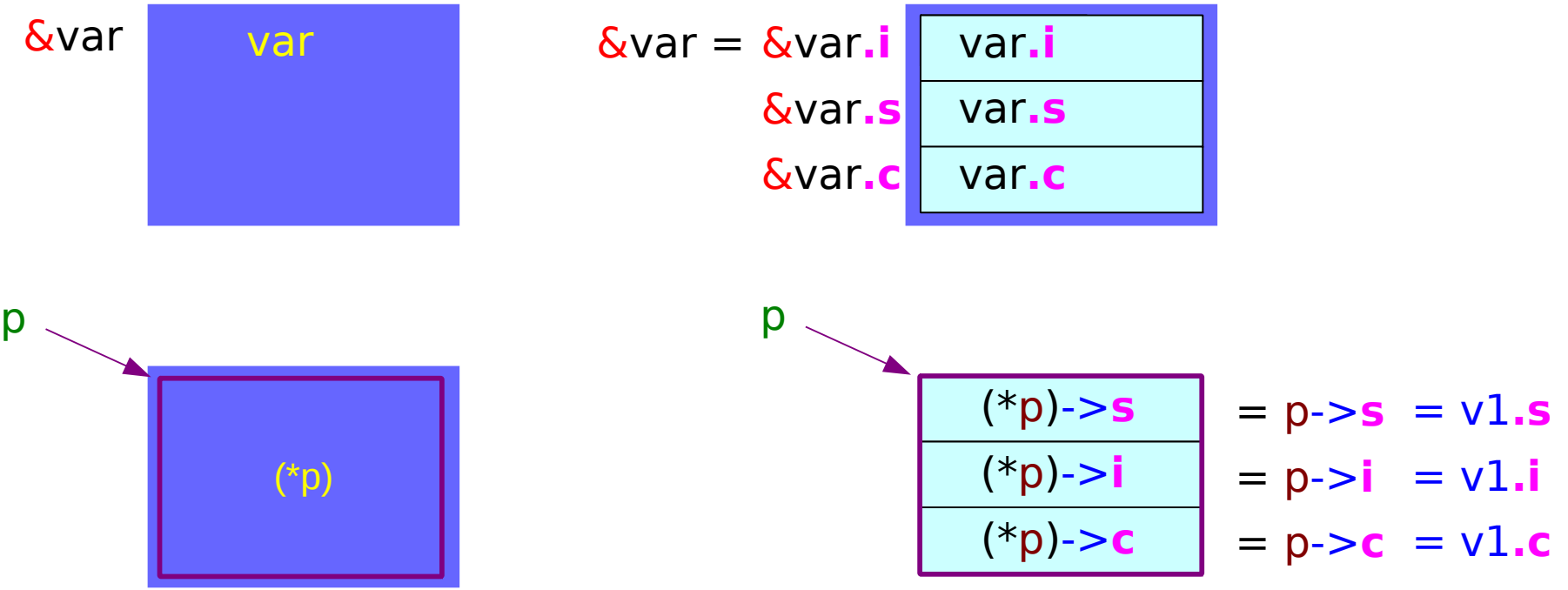
pointer variable assignment

```
p = &v1;
```



Dereferencing Pointers to Structures (1)

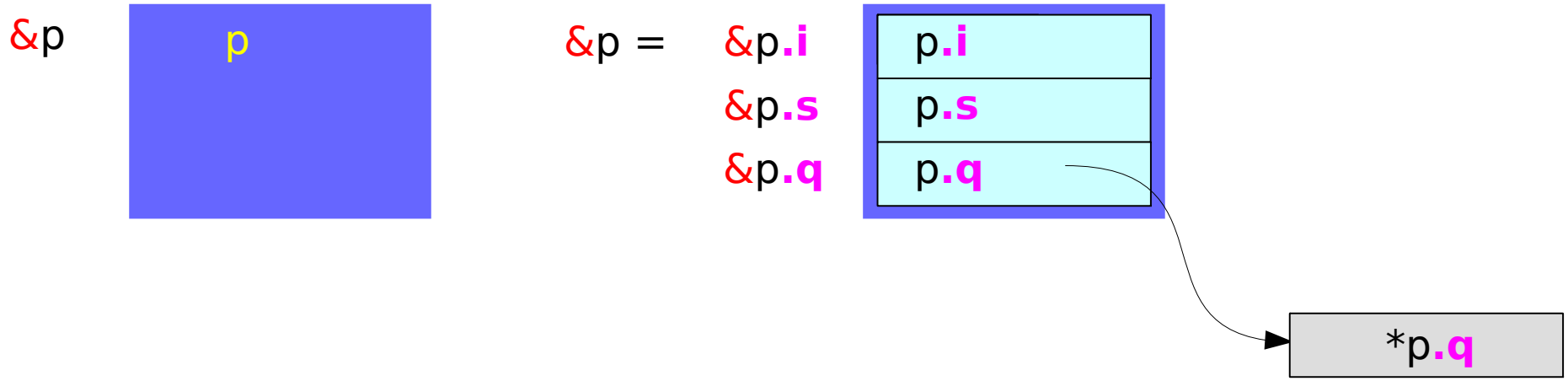
(*p).mem ⟺ p->mem



Dereferencing Pointers to Structures (2)

*p.mem ⟺ *(p.mem)

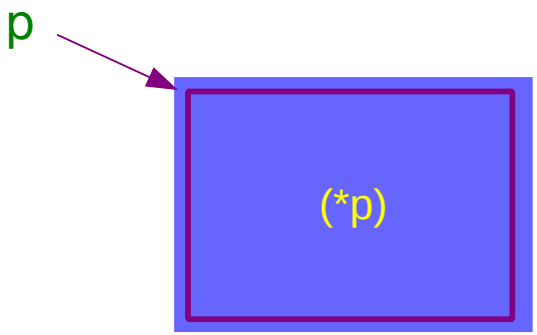
p : a struct variable
mem : another pointer variable

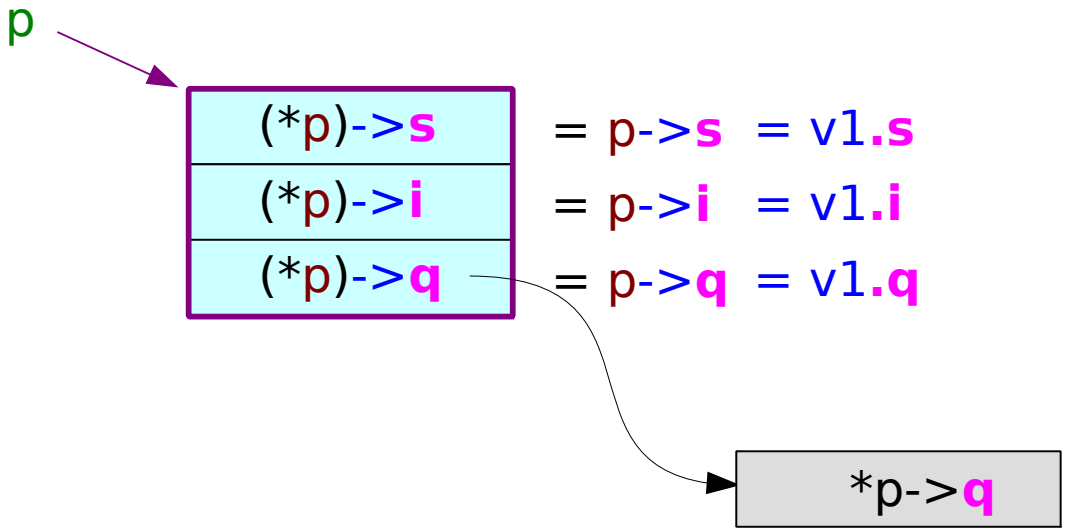


Dereferencing Pointers to Structures (3)

*p->mem ⟺ *(p->mem)

p : a pointer to a struct variable
mem: another pointer variable





Dereferencing Pointers to Structures

structure type

```
struct aaa {
        short       s;
        int         i;
        char        c;
} ;
```

definition

```
struct aaa    v1 = { 100, 10, 'A'} ;
```

```
struct aaa    *p = &v1;
```

(*p).mem

p->mem

*p.mem ⟺ *(p.mem)

*p->mem ⟺ *(p->mem)

Memory Layout (1)

structure type

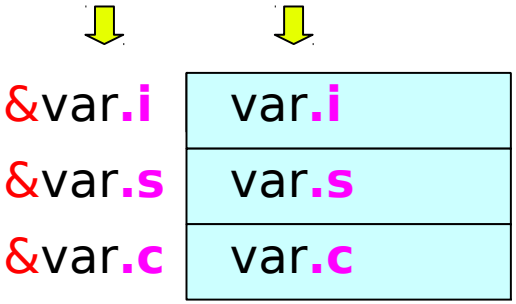
```
struct aaa {
        int         i;
        short       s;
        char        c;
} ;
```

definition

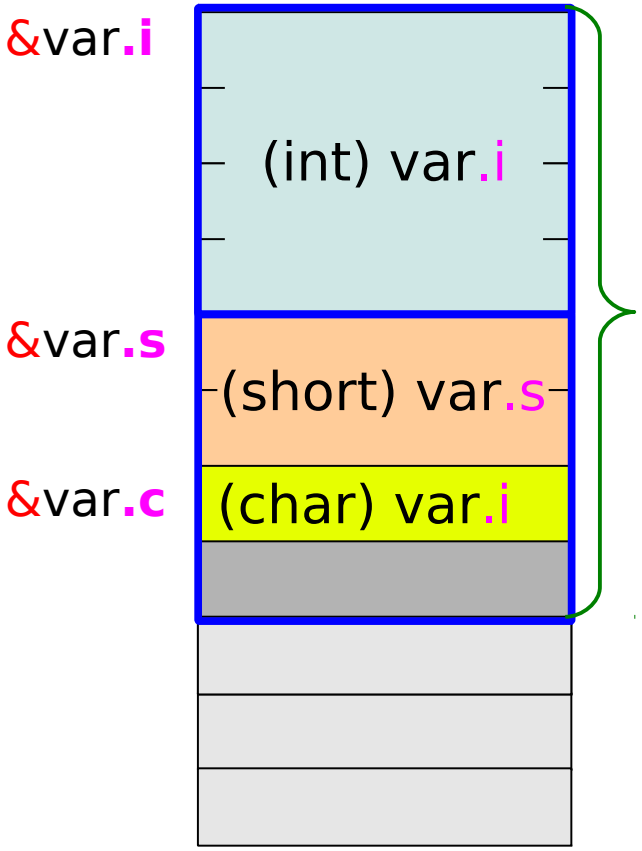
```
struct aaa   var;
```

var declaration

address data



sizeof (int) = 4 bytes
sizeof (short) = 2 bytes
sizeof (char) = 1 bytes



sizeof (var) = 8 bytes

Memory Layout (2)

structure type

```
struct aaa {
        short       s;
        int         i;
        char        c;
} ;
```

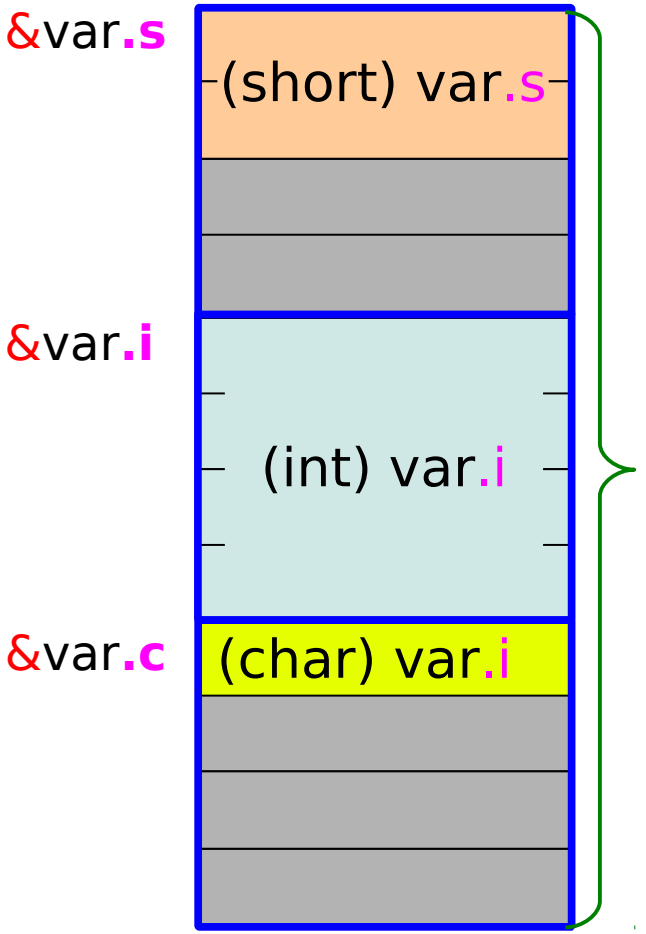
definition

```
struct aaa   var;
```

var declaration

address data

&var.s	var.s
&var.i	var.i
&var.c	var.c



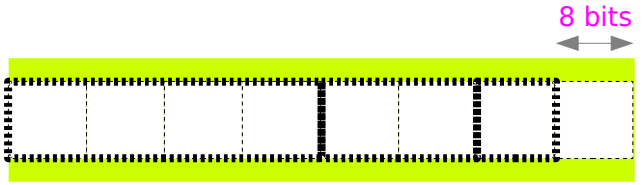
sizeof (var) = 12 bytes

Struct Definition and Declarations

structure type

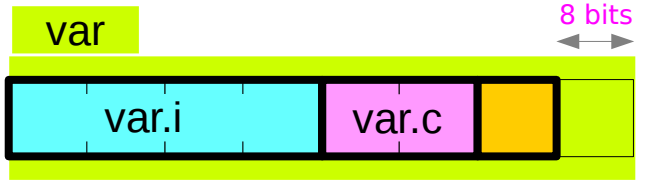
```
struct aaa {
        int         i;
        short       s;
        char        c;
} ;
```

definition



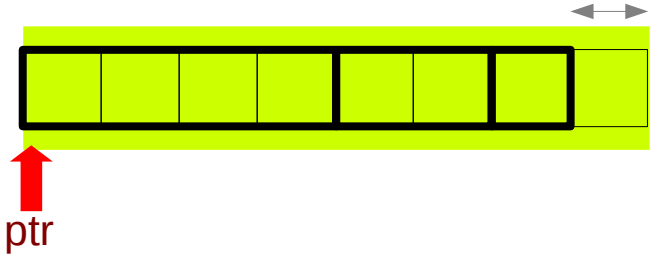

```
struct aaa    var;
```

var declaration




```
struct aaa    *ptr;
```

var declaration



Nested Structures

```
struct point {
        int   x ;
        int   y ;
} ;
```

```
struct rect {
    struct point   p1 ;
    struct point   p2 ;
} ;
```

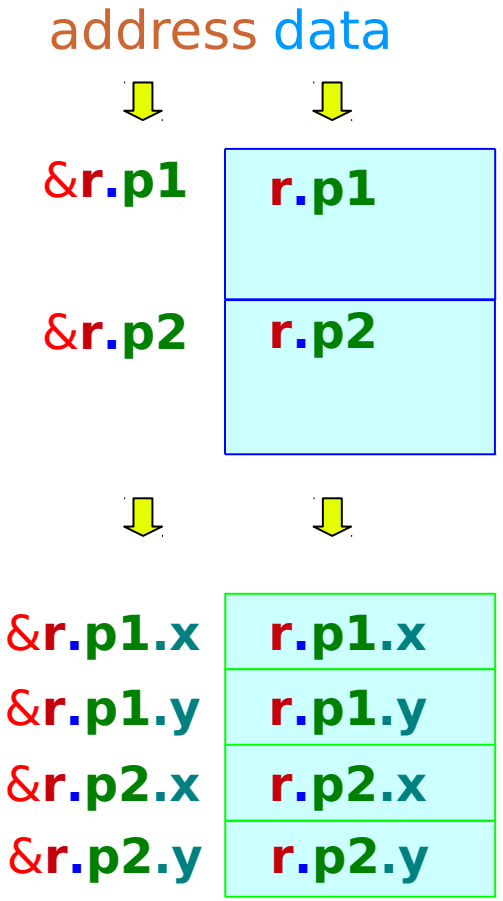
```
struct point   pa ;
```

```
struct point   pb ;
```

```
struct rect   r ;
```

```
pa.x   = 10 ;
pa.y   = 20 ;
pb.x   = 300 ;
pb.y   = 400 ;
```

```
r.p1   = pa ;
r.p2   = pb ;
```



Array Implementation

```
// within main()

int i ;

char *          name[3]  =    { “John”, “Mary”, “Baker” };
int             stid[3]    =    { 201101, 201102, 201103 };
int             eng[3]     =    { 94, 85, 90 };
int             math[3]    =    { 88, 92, 98 };
double          gpa[3];


for (i=0; i<3; ++i)
   printf(“---------------------------\n”);
   printf(“name: %s \n”,   name[i] );
   printf(“stid:    %d \n”,   stid[i] );
   printf(“eng:   %d \n”,   eng[i] );
   printf(“math: %d \n”,   math[i] );
   printf(“gpa:   %f \n”,    (eng[i] + math[i]) / 2. );
}
```

Struct Implementation

```
// outside main()
struct srec {
   char *          name ;
   int             stid ;
   int             eng ;
   int             math ;
   double          gpa ;
} ;


// within main()
struct srec S[3] =    {     { “John”,  201101,  94,  88 },
                            { “Mary”, 201102,  85, 92 },
                            { “Baker”, 201103, 90, 98 }    };

for (i=0; i<3; ++i)
   printf(“---------------------------\n”);
   printf(“name: %s \n”,    S[i].name );
   printf(“stid:   %d \n”,    S[i].stid );
   printf(“eng:   %d \n”,    S[i].eng );
   printf(“math: %d \n”,    S[i].math );
   printf(“gpa:   %f \n”,    (S[i].eng + S[i].math) / 2. );
}
```

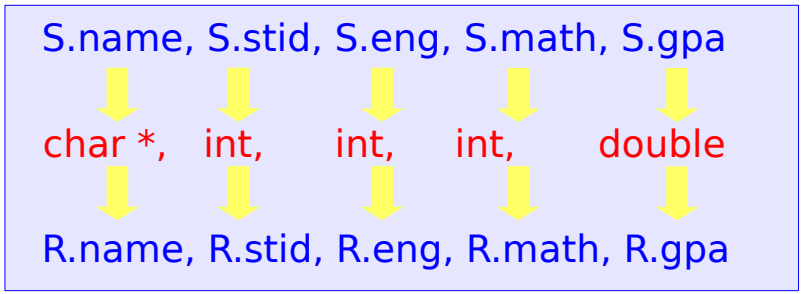
Struct as a function argument

```
// outside main()
struct srec {
   char *          name ;
   int             stid ;
   int             eng ;
   int             math ;
   double          gpa ;
} ;


double compute_gpa (struct srec R)
{
    return (R.eng + R.math) / 2. ;
}

// within main()
struct srec S =
            { “John”,  201101,  94,  88 };

double gpa;

gpa = compute_gpa ( S );
```

```
gpa = compute_gpa ( S );
```

```
double compute_gpa (struct srec R)
```

```
gpa = compute_gpa ( S );
```




```
double compute_gpa (struct srec R)
```

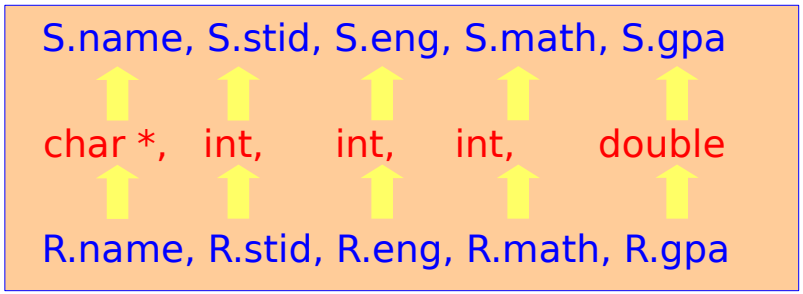
Struct returning function

```
// outside main()
struct srec {
   char *          name ;
   int             stid ;
   int             eng ;
   int             math ;
   double          gpa ;
} ;


struct srec  calc_gpa (struct srec R)
{
    R.gpa =  (R.eng + R.math) / 2. ;
    return R;
}

// within main()
struct srec S =
              { “John”,  201101,  94,  88 };

double gpa;

S = calc_gpa ( S );
```

gpa = calc_gpa (S);

struct srec calc_gpa (struct srec R)

S = calc_gpa (S);



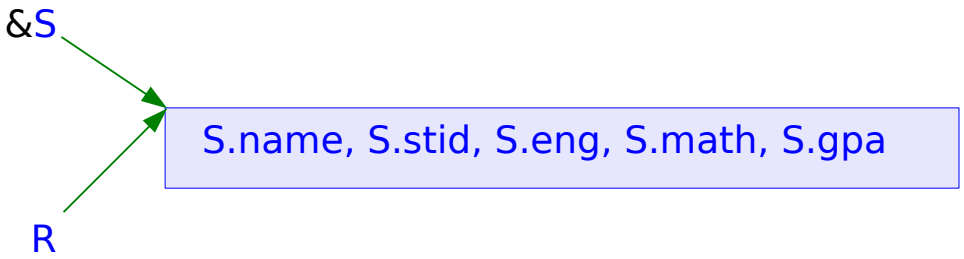
struct srec calc_gpa (struct srec R)

Struct pointer as a function argument

```
// outside main()
struct srec {
  char *          name ;
  int             stid ;
  int             eng ;
  int             math ;
  double          gpa ;
} ;


void fill_gpa (struct srec * R) {
  (*R).gpa =
        ((*R).eng + (*R).math) / 2. ;
}

// within main()
struct srec S =
            { "John",  201101,  94,  88 };

double gpa;

gpa = fill_gpa ( &S );
```

```
gpa = fill_gpa ( &S );
```

```
void fill_gpa (struct srec * R) ;
```




```
gpa = fill_gpa ( &S );
```



```
void fill_gpa (struct srec * R) ;
```

(*R).eng	⟺	R->eng
(*R).math		R->math
(*R).gpa		R->gpa

References

[1] Essential C, Nick Parlante
[2] Efficient C Programming, Mark A. Weiss
[3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
[4] C Language Express, I. K. Chun