

# Structure (1A)

---

Copyright (c) 2010 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to [youngwlim@hotmail.com](mailto:youngwlim@hotmail.com).

This document was produced by using OpenOffice.

# Struct Declaration (1)

structure type

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

```
struct name var;
```

structure type

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

# Struct Declaration (2)

structure type

```
struct name {
```

```
    int    i;
```

```
    short  s;
```

```
    char   c;
```

```
};
```

```
typedef struct name  stname_t ;
```

```
stname_t  var;
```

structure type

```
typedef struct name {
```

```
    int    i;
```

```
    short  s;
```

```
    char   c;
```

```
} stname_t;
```

```
stname_t  var;
```

# Struct Declaration (3)

structure type

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

```
struct name var;
```

structure type

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

```
typedef struct name stname_t ;
```

```
stname_t var;
```

structure type

```
typedef struct name {  
    int    i;  
    short  s;  
    char   c;  
} stname_t;
```

```
stname_t var;
```

structure type

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

# Struct Declaration

new type



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

# Typedef (1)

new type



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

```
typedef struct name  stname_t ;
```

# Typedef (2)

new type



```
typedef struct name {  
    int    i;  
    short  s;  
    char   c;  
} stname_t ;
```



# Struct Variable Declaration (1)

new type



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

```
struct name var;
```

# Struct Variable Declaration (2)

new type



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

# Struct Variable Declaration (3)

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

```
typedef struct name  stname_t
```

```
stname_t  var ;
```

# Array Implementation

```
// within main()

int i ;

char *      name[3] = { "John", "Mary", "Baker" };
int         std[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("std:   %d \n", std[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      stdid ;
    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.name[i] );
    printf("stdid:  %d \n", S.stdid[i] );
    printf("eng:   %d \n", S.eng[i] );
    printf("math:  %d \n", S.math[i] );
    printf("gpa:   %f \n", (S.eng[i] + S.math[i]) / 2. );
}
```

# Struct as a function argument

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

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

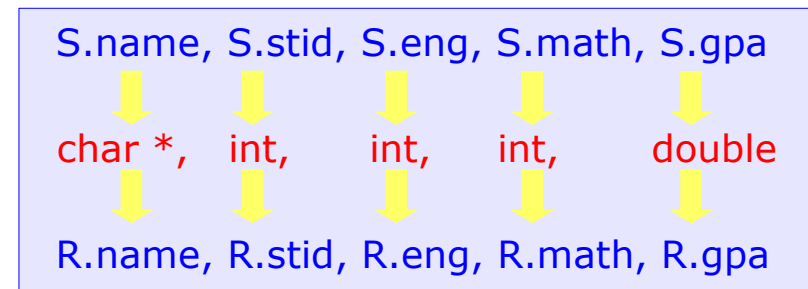
// within main()
struct srec S =
    { "John", 201101, 94, 88 };

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

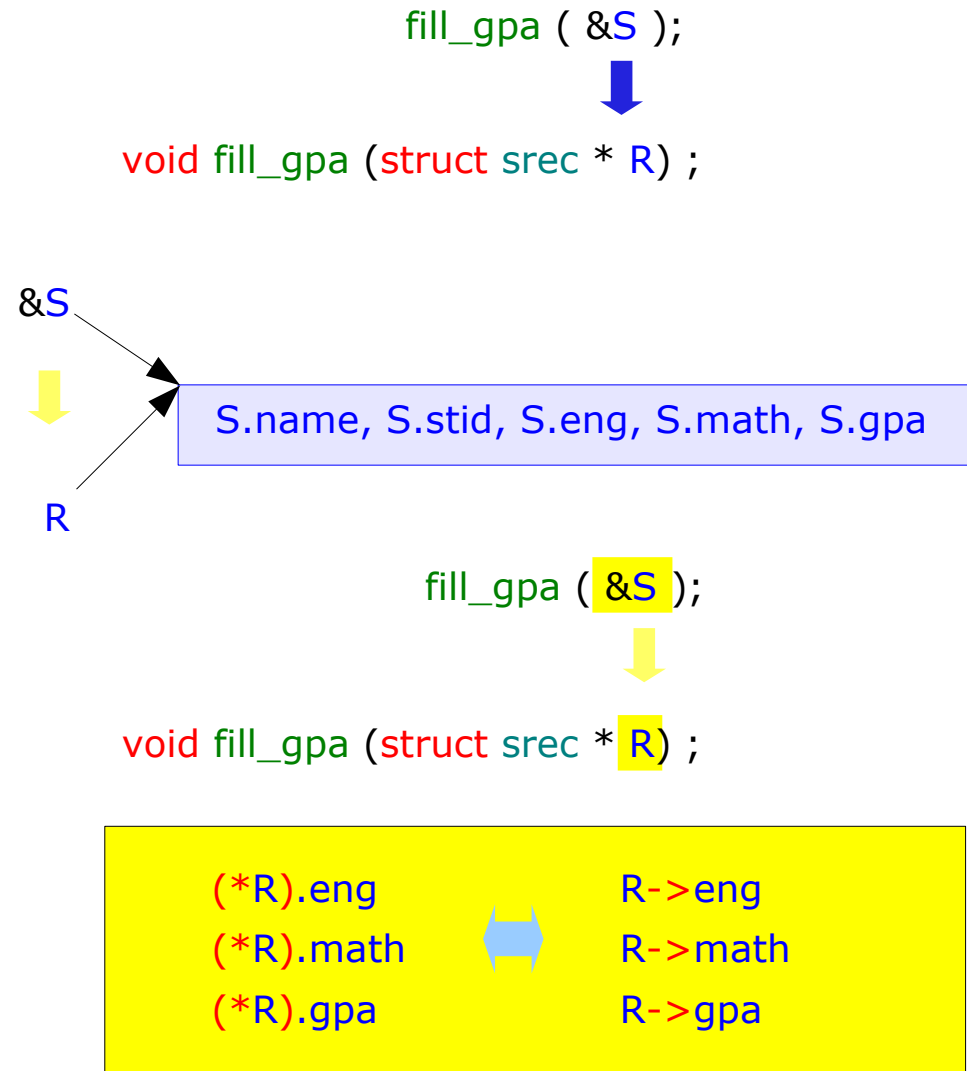
```
// outside main()
struct srec {
    char *   name ;
    int      stdid ;
    int      eng ;
    int      math ;
    double   gpa ;
};

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

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

double gpa;

fill_gpa ( &S );
```





## 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