

Applications of Structures (1A)

Copyright (c) 2009 - 2017 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.

This document was produced by using LibreOffice.

An Array of Structures
Incomplete Definition of Structures
Self-Referential Structures

Defining an array of a structure

Student ID Korean English Math Average

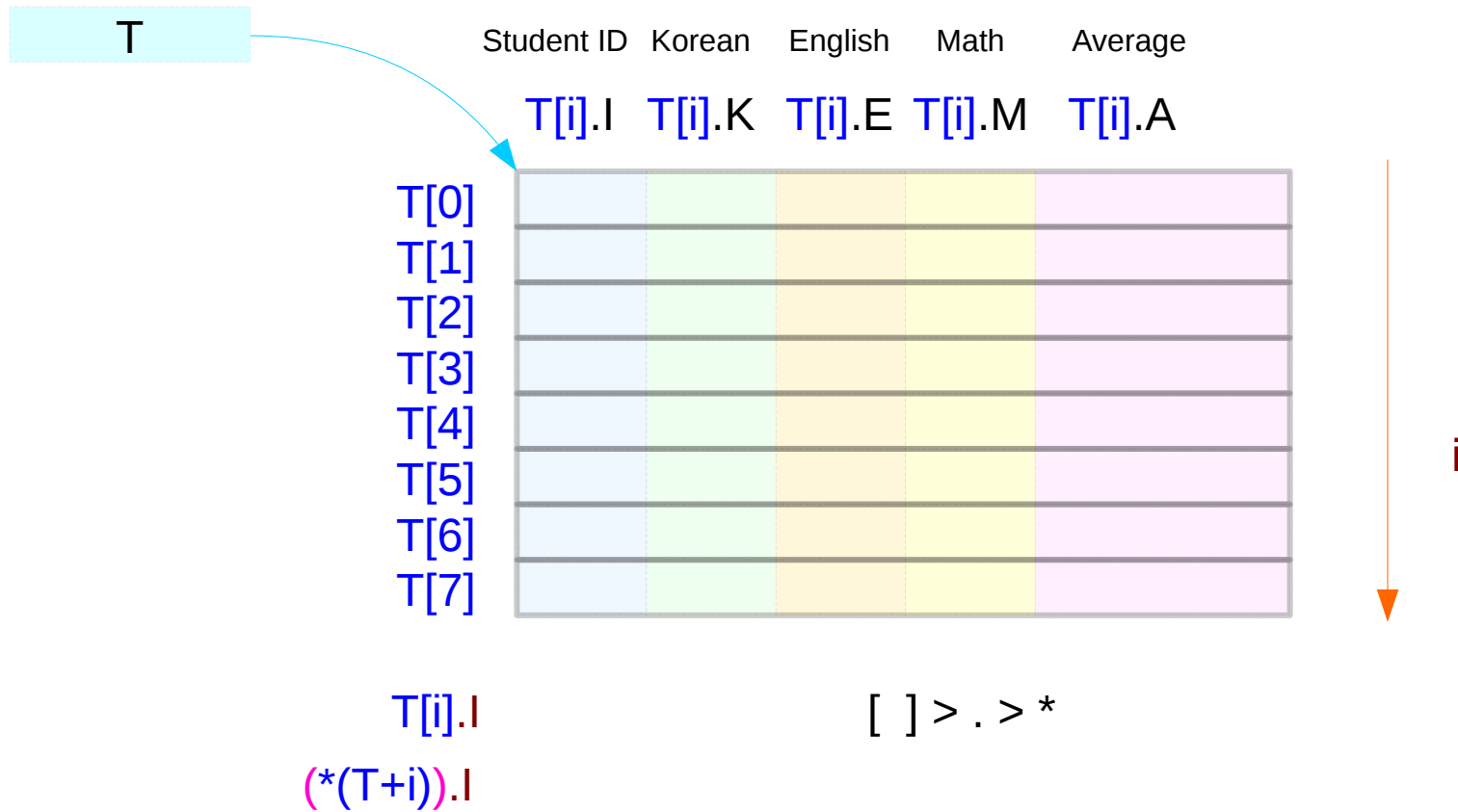
$S[i].I$ $S[i].K$ $S[i].E$ $S[i].M$ $S[i].A$

$S[0]$					
$S[1]$					
$S[2]$					
$S[3]$					
$S[4]$					
$S[5]$					
$S[6]$					
$S[7]$					

```
struct Stype {  
    int I;  
    int K;  
    int E;  
    int M;  
    double A;  
};
```

```
struct Stype S[SIZE];
```

Pointer to a Structure Array




struct Stype T[] : formal parameter

```
struct Stype { int I; int K; int E; int M; double A; };
struct Stype S[SIZE];
```

Array Name
Starting Address

S



```
void init_arrays (struct Stype T[]);
void pr_table (struct Stype T[]);
void pr_sorted_table (struct Stype T[]);
double Avg (struct Stype T[], int n);
void pr_averages (struct Stype T[]);
```

Struct Variable Declaration Summary

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

```
struct aaa var;
```

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

```
typedef struct aaa ATYPE ;
```

```
ATYPE var;
```

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

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

```
ATYPE var;
```

Incomplete Definition Range

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

At all these points, the structure type definition is **incomplete**

From here, the definition complete.

Incomplete Definition Usages

```
struct bbb { int i; struct bbb *s; };
```

incomplete definition
can be used with pointer variable declaration

```
typedef struct ccc Ctype;
```

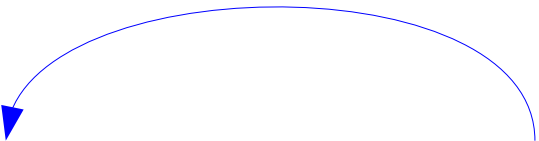
incomplete definition
can be used with **typedef**

```
void func( struct ddd *Ctype );
```

incomplete definition
can be used with pointer parameter declaration

Self-Referential Structures

```
struct bbb { int i; struct bbb *s; };
```



incomplete definition
can be used with pointer variable declaration

Incomplete Structure Definitions

structure type

```
struct aaa {  
  int      data;  
  struct aaa *next;  
} ;
```

```
struct aaa var;
```

structure type

```
struct aaa {  
  int      data;  
  struct aaa *next;  
};
```

```
typedef struct aaa ATYPE ;
```

```
ATYPE var;
```

structure type

```
struct aaa {  
  int      data;  
  struct aaa *next;  
} var ;
```

structure type

```
typedef struct aaa {  
  int      data;  
  struct aaa *next;  
} ATYPE ;
```

```
ATYPE var;
```

Changing Orders

```
struct aaa var;
```

structure type

```
struct aaa {  
    int data;  
    struct aaa *next;  
};
```

```
typedef struct aaa ATYPE ;
```

structure type

```
struct aaa {  
    int data;  
    struct aaa *next;  
};
```

```
ATYPE var;
```

structure type

```
struct aaa {  
    int data;  
    struct aaa *next;  
} var;
```

structure type

```
typedef struct aaa {  
    int data;  
    struct aaa *next;  
} ATYPE ;
```

```
ATYPE var;
```

Changing Tag Names

```
struct Atype var;
```

structure type

```
struct Atype {  
int data;  
struct Atype *next;  
};
```

```
typedef struct Atype Atype ;
```

structure type

```
struct Atype {  
int data;  
struct Atype *next;  
};
```

```
Atype var;
```

structure type

```
struct Atype {  
int data;  
struct Atype *next;  
} var;
```

structure type

```
typedef struct Atype {  
int data;  
struct Atype *next;  
} Atype ;
```

```
Atype var;
```

Types in the List Data Structures

```
typedef struct node node ;
```

```
node var;
```

structure type

```
struct node {  
    int data;  
    node *next;  
};
```

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
- [5] <https://pdos.csail.mit.edu/6.828/2008/readings/pointers.pdf>