

STRUCTURES

Dr. H. IREM TURKMEN

Structures

Arrays are good for dealing with groups of identically typed variables

But they are unsatisfactory for managing groups of differently typed data.

To service groups of mixed data, you need to use an aggregate type called a *structure*.

Structures

- A structure is like an array except that each element can have a different data type. Moreover, elements in a structure have names instead of index values.
- Without structures, a single person's record would be declared as:

```
char name[20], cID[11];
```

```
short day, month, year;
```

- What about multiple people's records?

```
char name[1000][20], cno[1000][11];
```

```
short day[1000], month[1000], year[1000];
```

Three ways to define a structure

// Without a TAG name

```
// Define the template and variable together WITHOUT a tag
// Have to create your variables where you define the structure
struct
{
    char ps_name[20], ps_cno[11];
    short ps_day, ps_month, ps_year;
} ps;
```

//Using a typedef name

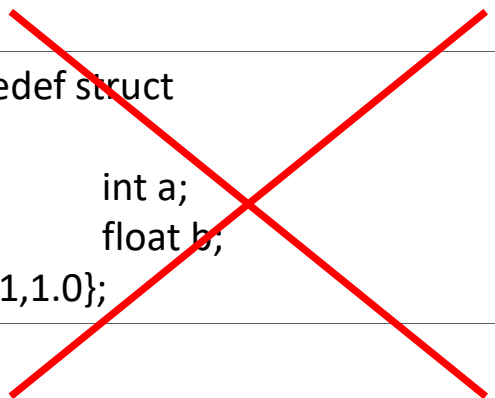
```
typedef struct
{
    char ps_name[20], ps_cno[11];
    short ps_day, ps_month, ps_year;
} PERSONALSTAT;
// Declare a variable from above template by;
PERSONALSTAT ps;
```

//Using a TAG name

```
// Define the template and variable together WITH a tag
struct personalstat
{
    char ps_name[20], ps_cno[11];
    short ps_day, ps_month, ps_year;
} ps, psarr[1000], *ptrps;
// Declare a variable from above template by;
struct personalstat ps;
struct personalstat psarr[1000], *ptrps;
```

Initialization

- PERSONALSTAT ps = { "George Smith", "002340671", 3, 5, 1946 };
- Initializer is not allowed in typedef



```
typedef struct
{
    int a;
    float b;
}s={1,1.0};
```

```
struct
{
    char ps_name[20], ps_cno[11];
    short ps_day, ps_month, ps_year;
} ps={ "George Smith", "002340671", 3, 5, 1946 };
```

```
struct personalstat
{
    char ps_name[20], ps_cno[11];
    short ps_day, ps_month, ps_year;
} ps = { "George Smith", "002340671", 3, 5, 1946 };
```

Referencing Structure Members & Arrays

```
typedef struct
{
    char ps_name[20], ps_no[11];
    short ps_day, ps_month, ps_year;
} PERSONALSTAT;
int main()
{
    PERSONALSTAT ps;           //a variable
    PERSONALSTAT psArray[10];  //an array
    PERSONALSTAT *psPtr;       //a pointer
    ps.ps_day=20;
    psArray[0]. ps_day=20;
    psPtr=&psArray[0];
    (*psPtr).ps_day=20;
    psPtr ->ps_day=20;
}
```

```
struct PERSONALSTAT
{
    char ps_name[20], ps_cno[11];
    short ps_day, ps_month, ps_year;
};
int main()
{
    struct PERSONALSTAT ps;
    struct PERSONALSTAT psArray[10];
    struct PERSONALSTAT *psPtr;
    ps.ps_day=20;
    psArray[0]. ps_day=20;
    psPtr=&psArray[0];
    (*psPtr).ps_day=20;
    psPtr ->ps_day=20;
}
```

Nested Structures

```
typedef struct
{
    char day;
    char month;
    short year;
} DATE;
typedef struct
{
    char ps_name[20], ps_tcno[11];
    DATE ps_birth_date;
} PERSONALSTAT;
// Declare an array from above definition:
PERSONALSTAT psarr[1000];
psarr[j].ps_birth_date.day=25;
```