structs in C

structs allow a programmer to group different types of data into a single entity.

```c
struct personType
{
    char name[20];
    char rank[10];
    int serno;
};
```

The above statement is the declaration of a new type! It does NOT declare any storage!

The individual components of the struct are called members.

A struct Example

To declare a struct "variable":

```c
personType soldier;
```

To refer to members of the struct:

```c
cin >> soldier.name >> soldier.rank >> soldier.serno;
```
structs with Array Members

Members can be any type, including arrays and other structs.

```c
struct studenttype
{
    int studid;
    int tests[3];
    int quizzes[3];
    int final;
    char coursegrade;
};
```

structs with Arrays

Some examples using arrays within structs

```c
 studenttype student;

 for (i = 0; i < 3; i++)
    student.quizzes[i] = 0;

 testavg = (student.tests[0] + student.tests[1])
       / student.tests[2];

 or

 testavg = 0;
 for (i = 0; i < 3; i++)
    testavg += student.tests[i];

 testavg = testavg / 3.0;
```
Arrays of structs

`studenttype allstudents[50];`

`for (i = 0; i < 50; i++)`
`    for (j = 0; j < 5; j++)`
`        allstudents[i].quizzes[j] = 0;`

`for (i = 0; i < 50; i++)`
`{`
`    cin >> allstudents[i].studid;`
`    for (j = 0; j < 5; j++)`
`        cin >> allstudents[i].tests[j];`
`    for (j = 0; j < 5; j++)`
`        cin >> allstudents[i].quizzes[j];`
`    cin >> allstudents[i].final;`
`}`

structs and Functions

`struct examtype`
`{`
`    int a, b, c;`
`    float x, y;`
`};`
`void func(examtype x)`
`{`
`    int sumints;`
`    sumints = x.a + x.b + x.c;`
`    return sumints;`
`} // END func`

`int main`
`{`
`    examtype arg;`
`    int r;`
`    r = func(arg);`
`} // END main`