C structures

September 25, 2019 7:57 AM

Announcements

- Quiz next week: Srings, string functions

```
struct Employee {
  int empnum;
  char name[MAXLEN];
  double salary;
};
```

Records (structures)

- need to deal with related data
- strucure tells compiler how it is laid out in memory and details the attirbute names
- memory is allocated when strucre var is declared

    struct Employee boss1;

Initialisation and access

- Constant values can be assigned to members of a structure when structure var is declared

    Employee sessional_a= {4314324, "Geoff", 23232.27}

Arrays of structures

- syntax for declaring and accessing is the same as arrays for any other type
- can be declared into dynamic memory

    Employee vice_president;*

    vice_president = (Employee) malloc(sizeof(Employee));*
- To access members in dynamic memory, deference the pointer to get the structure

    *(*vice_president).salary += 10000.00;*

    OR *vice_president -> salary = 150000.00;*
- Dynamic Arrays of structures

```
Employee* staff_senior;
staff_senior = (Employee*) malloc(num_staff_senior *
                                  sizeof(Employee));

// accessing using array syntax
staff_senior[i].empnum = 100 + i;

// accessing using pointer arithmetic
(staff_senior + i)->salary = 80000;
(*(staff_senior + i)).salary *= 1.05;
```

Nested structures

typedef struct{

    int empnum;

    char name[MAXLEN];

    double salary;

    Date dob;

} Employee;

- Passing structs by parameters

Structure variables

- size of the employee structure given that in= 4, char*=4 and double = 8
- = 16 bytes for a struct using each type once