



# Structures in C

Atul Gupta



# Structures

- A group of meaningful, related data to identify an entity

```
main() {  
    struct book{  
        char accession_no[15];  
        char *title;  
        char[] *authors;  
        float price;  
        ...  
    }  
    //structure declarations  
    struct book book1,book2,book3;  
    struct book textbooks[10000];  
    struct book *book_ptr;  
}
```

```
main() {  
    printf("Book1 price is %f", book1.price);  
    printf("Book1 price is %f", book_ptr->price);  
}
```



## 'struct' in C: An Example

```
#include <stdio.h>
#include <string.h>
#include <conio.h>

struct Employee{
    /* the structure type */
    char lname[20];      /* last name */
    char fname[20];     /* first name */
    int age;             /* age */
    float wages;        /* e.g. 12.75 per hour */
};

struct Employee anEmployee; /* declare an structure */
void showName(struct Employee *p); /* function prototype */

void showName(struct Employee *e)
{
    printf("\n%s ", e->fname); /* p points to a structure */
    printf("%s ", e->lname);
    printf("%d\n", e->age);
}
```

```
int main(void)
{
    struct Employee *st_ptr; /* a pointer to a structure */
    st_ptr = &anEmployee; /* point the pointer to my_struct */
    strcpy(anEmployee.lname,"Jensen");
    strcpy(anEmployee.fname,"Ted");
    printf("\n%s ",anEmployee.fname);
    printf("%s\n",anEmployee.lname);
    getch();
    anEmployee.age = 63;
    showName(st_ptr); /* pass the pointer */
    getch();
    return 0;
}
```



## Array of Structures: Example

```
#define MAX 100

struct addr {
    char name[30];
    char street[40];
    char city[20];
    char state[3];
    unsigned long int zip;
} addr_list[MAX];

//The functions to be used
//with the collection
void init_list();
void enter_address();
void delete_address();
void list();
int menu_select();
int find_free();
```

```
int menu_select() {
    char s[80];
    int c;
    printf("\n\n 1. Enter a name\n");
    printf(" 2. Delete a name\n");
    printf(" 3. List the file\n");
    printf(" 4. Quit\n");
    do {
        printf("\nEnter your choice: ");
        gets(s);
        c = atoi(s);
    } while(c<0 || c>4);
    return c;
}

/* Find an unused structure. */
int find_free()
{
    int i;

    for(i=0; addr_list[i].name[0] && i<MAX; ++i) ;

    if(i==MAX) return -1; /* no slots free */
    return i;
}
```



## Array of Structures: Example cont...

```

/* Insert an address */
void enter_address() {
    int slot; char s[80];

    slot = find_free();

    if(slot==-1) {
        printf("\nList Full"); return;
    }

    printf("Enter name: ");
    gets(addr_list[slot].name);

    printf("Enter street: ");
    gets(addr_list[slot].street);

    printf("Enter city: ");
    gets(addr_list[slot].city);

    printf("Enter state: ");
    gets(addr_list[slot].state);

    printf("Enter zip: ");
    gets(s);
    addr_list[slot].zip = strtoul(s, '\0', 10);
}

```

```

/* Delete an address. */
void delete_address()
{
    int slot;
    char s[80];

    printf("enter record #: ");
    gets(s);
    slot = atoi(s);

    if(slot>=0 && slot < MAX)
        addr_list[slot].name[0] = '\0';
}

```

```

/* Display the list on the screen. */
void list()
{
    int i;

    for(i=0; i<MAX; ++i) {
        if(addr_list[i].name[0]) {
            printf("%s\n", addr_list[i].name);
            printf("%s\n", addr_list[i].street);
            printf("%s\n", addr_list[i].city);
            printf("%s\n", addr_list[i].state);
            printf("%lu\n\n", addr_list[i].zip);
        }
    }
}

```



## Array of Structures: Example cont...

```

void init_list()
{
    int i;
    for(i=0; i<MAX; ++i)
        addr_list[i].name[0] = '\0';
}

```

```

int main()
{
    char choice;

    init_list(); /* initialize the structure array */
    for(;;) {
        choice = menu_select();
        switch(choice) {
            case 1: enter_address();
                break;
            case 2: delete_address();
                break;
            case 3: list();
                break;
            case 4: exit(0);
        }
    }
    return 0;
}

```



## Structure of Structures