**Pointer to structures**

If you want a pointer to a structure you have to use the -> (infix operator) instead of a dot.  
Take a look at the following example:

#include<stdio.h>

typedef struct telephone

{

char \*name;

int number;

}TELEPHONE;

int main()

{

TELEPHONE index;

TELEPHONE \*ptr\_myindex;

ptr\_myindex = &index;

ptr\_myindex->name = "Jane Doe";

ptr\_myindex->number = 12345;

printf("Name: %s\n", ptr\_myindex->name);

printf("Telephone number: %d\n", ptr\_myindex->number);

return 0;

}

**Note:** The -> (infix operator) is also used in the printf statement.

**Unions**

A union is like a structure in which **all members** are stored at the **same** address. Members of a union can only be accessed one at a time. The union data type was invented to prevent memory fragmentation. The union data type prevents fragmentation by creating a standard size for certain data. Just like with structures, the members of unions can be accessed with the . and -> operators. Take a look at the example:

#include<stdio.h>

typedef union myunion

{

double PI;

int B;

}MYUNION;

int main()

{

MYUNION numbers;

numbers.PI = 3.14;

numbers.B = 50;

return 0;

}