Strucutres in C

Basics of Structures

A **structure** is a collection of one or more variables, possibly of different types, group together under a single name. Structures permit a group of related variables to be treated as a unit instead of separate entities.

A structure for a 2D-point in space. The Two components can be placed in a structure declared like this:

```
struct point {
        int x;
        int y;
};
```

The keyword `struct` introduces a structure declaration, which is a list of declarations enclosed in braces. An optional name called a **structure tag** (`point` is the tag above) may follow the word `struct`. The tag names this kind of structure, and can be used subsequently as a shorthand for the part of the declaration in braces.

The variables named in a strucute are called **members**. A structure member or tag and an ordinary (non-member) variable can have the same name without conflict.

A `struct` declaration defines a type. The right brace that terminates the list of members may be followed by a list of variables, just as any basic type. That is,

```
struct { ··· } x, y, z;
```

If the strucute declaration is tagged, the tag can be used later in definitions of instances of the strucute.

```
sturct point pt;
```

defines a variable `pt` which is a structure of type `struct point`. A structure can be initialized by following its definition with a list of initializers, each a constant expression, for members:

```
struct point maxpt = { 320, 200 };
```

A member of a particular structure is referred to in an expression by a construction of the form

```
structure-name.member
```

The structure member operator '.' connectes the structure name and the member name.

Structures and Functions

The only legal operations on a structure are copying it or assigning to it as a unit, taking its address with &, and accessing its members. Copy and assignment include passing arguments to functions and returning values from functions. Structures may not be compared. A structure may be initialized by a list of constant member values; an automatic structure may also be initialized by an assignment.

The declaration

```
struct point *pp;
```

says that `pp` is a pointer to a structure of type `struct point`. If `pp` points to a `point` structure, `*pp` is the structure, and `(*pp).x` and `(*pp).y` are the members.

If `p` is a pointer to a strucute, then

```
p -> member-of-structure
```

refers to a particular member.