

2.3.2 Unions

- A union declaration is similar to a structure, but the fields of a union must share their memory space. This means that only one field of the union is active at any given time.
- For example, to add different fields for males and females, we would change our definition of humanBeing to:

```
typedef struct {  
    enum tagField {female, male} sex;  
    union {  
        int children;  
        int beard;  
    } u;  
} sexType;  
typedef struct {  
    char name[10];  
    int age; float salary; date dob; sexType sexInfo;  
} humanBeing;
```

humanBeing person1, person2;

We could assign ~~to~~ values to person1 and person2 as:

person1.sexInfo.sex = male; person1.sexInfo.u.beard = FALSE;
and person2.sexInfo.sex = female; person2.sexInfo.u.children = 4;

- The value in the tag field allows us to determine which field in the union is active. C does not verify that we use the apt field.

2.3.4 Self-Referential Structures

- A self-referential structure is one in which one or more of its components is a pointer to itself. Self-referential structures usually require dynamic storage management routines (malloc and free) to explicitly obtain and release memory.