**Lecture 10 Unions and Enums**

**Unions**

* A union is a UDT that can hold any type of variables of a given set in the same memory block, so as to share space. The size of a union is the maximum size of its types
* A union type variable can hold different types of data but can only hold one type at a time

Syntax:

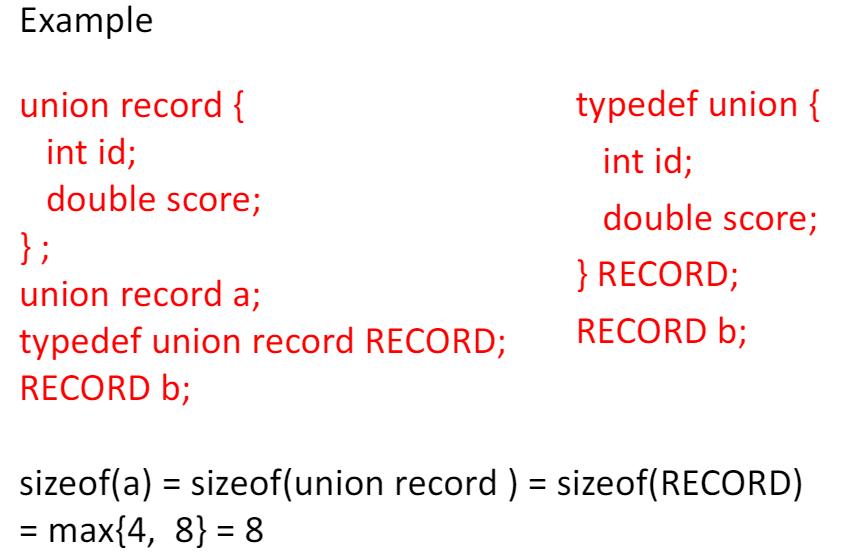
Union union\_name{

Data\_type1 var-name1;

Data\_type2 var-name;

};

* The syntax of using unions is similar to using structures



**Accessing union elements**

Example

RECORD a;

a.id = 1;

int rid = a.id;

printf(“%d”, rid);//print 1

a.score = 88.0;

printf(“%f”, a.score); //print 88.0000

Example

RECORD b; //cannot be initialized like RECORD b =10;

b.id=10; //b holds integer 10

**Unions vs Structures**

* Each variable of a structure has its own memory space, so can hold all element data simultaneously
* Union uses the same memory location for different variable. That means its variables share the same memory space, so can only hold one variable data at a time
* Similar to structure type, union supports
  + Assignment operations
  + Pointer operations
  + Passing union to function copies union data to function local
  + Function return, copying function local union out of function

**Enumerations**

1. An enumeration is a UDT to represent integral constants by names
2. An enum type value is stored as int type value. The sizeof enum data type is equal to sizeof(int)
3. An enum type data is used in the same way as an int data type

Example

C does not have Boolean type. We can use enumeration to define a Boolean type, which has two symbols false (represented by 0) and true (represented by 1). As a result, we can use the term false and true instead of 0 and 1 in source code programming such making the code more readable.

Advantage over using #define is that enum is a data type, int operations can be applied to a numeration type.

**Syntax**

Enum enum\_name{

Const1, //or const1 = value1

Const2,

…,

ConstN

};

This tells the compiler to associate const1 to 0, const2 to 1,… The compiler associates next one by increasing current value by 1, except a default association value is given.

Enum Boolean{

False,

True

};

Remember to define false first so it gets the value of zero