Unions and Enum Types

- Example

```c
union Value
{
    char sc;    // signed char  1 bytes
    short ss;   //  2 bytes
    int i;      //  4 bytes
    unsigned char uc; //  1 byte
    unsigned int u;  //  4 bytes
    float f;     //  4 bytes
    double d;    //  8 bytes
    void *p;     //  4 bytes
};

Value v;
```
• Typical Application:
  – two or more variables
  – different types
  – not used at the same time

• Feature: space is allocated for the largest type, not for all members

• arrays, pointer, parameters, return types as for structs

• Processing done in an if or switch statement

• Used with a flag to choose processing path

Example:
int type=2; //type is a flag: 0= char, 1= short, and so on
switch (type)
{
    case 0:
        cout << endl << sizeof(char); break;
    case 1:
        cout << endl << sizeof(short); break;
    case 2:
        cout << endl << sizeof(int); break;
    case 3:
        cout << endl << sizeof(unsigned char); break;
    case 4:
        cout << endl << sizeof(unsigned int); break;
    case 5:
        cout << endl << sizeof(float); break;
    case 6:
        cout << endl << sizeof(double); break;
    case 7:
        cout << endl << sizeof(void *); break;
    default:
        cout << endl << "error";
}
Enumerated Types

- ordered subset of integers starting at 0.
- special rules for assignment and arithmetic
- can be compared
- can be used as array subscripts
Example:

```cpp
enum Type{CHAR, SHORT, INT, UCHAR, UINT, FLOAT, DOUBLE, VOIDPTR};

Type flag;
flag = UINT;

switch (flag)
{
    case CHAR:
        cout << endl << sizeof(char); break;
    case SHORT:
        cout << endl << sizeof(short); break;
    case INT:
        cout << endl << sizeof(int); break;
    case UCHAR:
        cout << endl << sizeof(unsigned char); break;
    case UINT:
        cout << endl << sizeof(unsigned int); break;
    case FLOAT:
        cout << endl << sizeof(float); break;
    case DOUBLE:
```
cout << endl << sizeof(double); break;
case VOIDPTR:
    cout << endl << sizeof(void *); break;
default:
    cout << endl << "error";
}