2-5類別、結構與聯合彼此相關

```
class complex {
} ;
struct complex {
} ;
union complex {
} ;
```

class vs. struct

```
class complex {
    int  a, b; // a+bi
public:
    set(int x, int y) ;
    print() ;
} ;
void main() {
    complex c ;
    c.set(5,3); c.print() ;
}
```

```
struct complex {
    set(int x, int y) ;
    print() ;
private:
    int a, b;
} ;
void main() {
    complex c ;
    c.set(5,3); c.print() ;
}
```

Class vs. Struct

- 異同
  - 都可以宣告資料成員(data members)與成員函數(member functions)
  - 預設的存取限制: private vs. public
- 討論
  - 既生struct ，何生class

Class vs. Struct

- 習慣

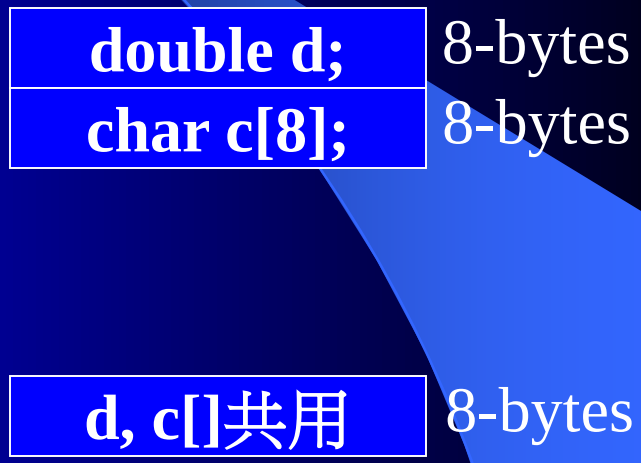
```
class complex {
    int a, b ;
public:
    set(int x, int y) ;
    …..
} ;
```

```
struct student {
        string name ;
        double GPA;
} ;
```

Union

- 多個變數共用一個記憶體區塊

```
struct sbits {
    double d ;
    char c[sizeof(double)] ;
};
union ubits {
    double d ;
    char c[sizeof(double)] ;
};
```



union vs. struct

```
void main() {
    sbits a ; //使用struct
    a.d = 314898;
    strcpy(a.c, “Hello”) ;
    ubits b ; //使用union
    b.d = 314898;
    strcpy(b.c, “Hello”) ;
}
```

d:314898	**c[]: ????**
d:314898	**c[]: “Hello”**
314898	
“Hello”	

匿名聯合(anonymous)

```
// 不必特別宣告變數
void main() {
    union fourbits {int i ; char ch[4]; } ;
    fourbits x ;
    x.i = 10; strcpy((char *)x.ch,”sam”) ;
    union { int i ; char ch[4];}; //匿名聯合
    i = 10 ;
    strcpy(ch, “sam”) ;
}
```

範例二: union也可以有成員函數

```
//自行run一次
union bits {
    bits(double n) ;
    void show_bits() ; //show 出每一個位元
    double d ;
    unsigned char c[sizeof(double)] ;
};
bits::bits(double n) { d = n; }
void bits::show_bits(){cout<<d<<c<<endl;}
……
```