**structure type structure type type**

pointer \* constant void

pointer array & variable num

reference array bool

reference array char

struct struct\_type struct

object class class

enum enum\_type enum

**⑴type**

**1.struct\_type**

definition:struct struct\_type\_name{struct\_body:data}

call:struct\_type\_name

**2.class**

definition:class class\_name{class\_body:data-attribute behavior-method}

class subclass\_name:derived\_way base\_class{subclass\_body:data-attribute behavior-method}

call:class\_name

**3.enum\_type**

definition:enum enum\_type\_name{enumerator=value}

call:enum\_type\_name

**⑵basic**

**1.constant**

definition:const data\_type const\_name=value

call:const\_name

**2.variable**

definition:data\_type var\_name=value

call:var\_name

**3.array**

definition:data\_type array\_name[index/key]={values}

call:array\_name[index/key]

**4.struct**

definition:struct\_type\_name struct\_name={arguments}

call:struct\_name.member\_name struct\_name->member\_name

**5.object**

definition:class\_name object\_name(constructor\_arguments)

call:object\_name.member\_name object\_name->member\_name

**6.struct array**

definition:struct\_type\_name array\_name[index/key]={arguments}

call:array\_name[index/key].member\_name array\_name[index/key]->member\_name

**7.object array**

definition:class\_name array\_name[index/key]={constructor\_arguments}

call:array\_name[index/key].member\_name array\_name[index/key]->member\_name

**8.enum**

definition:enum\_type\_name enum\_name

call:enum\_name=enumerator~value

**⑶advance**

**1.pointer**

definition:prefix\_type \* pointer\_name=address

call:pointer\_name \*pointer\_name

instance:constant pointer variable pointer array pointer string pointer struct pointer object pointer

**2.pointer array**

definition:prefix\_type \* array\_name[index/key]={address}

call---array\_name[index/key] \*array\_name[index/key]

**3.reference**

definition:prefix\_type & reference\_name=target

call:reference\_name

instance:constant reference variable reference array reference struct reference object reference

**⑷function**

**1.function**

definition:return\_type function\_name(input\_type parameters){function\_body}

call:function\_name(arguments)

**2.function pointer**

definition:return\_type \* pointer\_name(input\_type parameters)=function\_name

call:\*pointer\_name(arguments)

**3.function pointer array**

definition:return\_type \* pointer\_name[index/key](input\_type parameters)={function\_names}

call:\*pointer\_name[index/key](arguments)

**⑸template**

**1.function template**

definition:template <class parameters> parameters\_type function\_name(parameters\_type parameters\_parameters) {parameters\_function\_body}

call:function\_name<arguments>

**2.class template**

definition:template <class parameters> class class\_name{parameters\_class\_body}

call:class\_name<arguments>