**复合类型**

1. **引用（reference）P50**

**1.1引用注意**

（1）引用即别名，程序把引用和它的初始值绑定在一起，而非将初始值拷贝给引用；

（2）引用必须初始化，且初始值必须为一个对象，而不能为具体数字；

（3）初始的对象类型必须与定义的引用类型相同；

（4）因为引用本身不是一个对象，所以不能定义引用的引用。

1. **指针（pointer）P52**

**2.1指针和引用的异同点**

（1）和引用相同，指针也实现对其他对象的间接访问；

（2）指针本身就是一个对象，允许对指针赋值和拷贝；

（3）指针可以指向几个不同的对象；

（4）指针无须在定义时赋初值。

**2.2指针值（即地址&）有如下4种状态：**

（1）指向一个对象；

（2）指向紧邻对象所占空间的下一个位置；

（3）空指针，指针不指向任何对象；

（4）无效指针。

**2.3对指针解引用\*可得所指对象**

**2.4空指针，生成空指针的方法：**

（1）int \*p1 = nullptr;

（2）int \*p2 = 0;

（3）int \*p3 = NULL;

**Const指针（const pointer）**

指针是对象而引用不是，所以允许把指针定为常量。

常量指针必须初始化，之后它的值（存在在指针中的那个地址）就不能再改变了。

例：int num = 0;

int \*const p3 = &num; //可以修改其指向的对象

const double pi = 3.14;

const double \*const pip = &pi; //pip是指向双精度浮点型常量 的常量指针

**顶层const（top-level const）和底层const（low-level const）**

**顶层const：**指针本身是常量，也可以表示任意的对象是常量。

例：int i = 0;

int \*const p1 = &i;

const int ci = 41;

**底层const：**指针所指的对象是常量，与指针和引用等复合类型的基本类型部分相关。

例：const int ci = 41; //不能改变ci的值，ci是顶层const

const int \*p2 = &ci; //允许改变p2的值，p2是底层const

const int &r = ci; //用于声明引用的const都是底层const

**const int \*const p3 = p2; //p3既是顶层const，也是底层const**

当执行对象拷贝操作时，顶层const不受影响。而对底层const的限制不容忽视，拷入和拷出的对象必须具有相同的底层const资格，或者两个对象的数据类型必须能够转换。通常是非常量转换为常量。

例：int \*p = p3; //错，p3包含底层const，而p没有

p2 = p3;

p2 = &i; //? int\*可以转换为const int\*,究竟如何转换

int &r = ci; //错，普通int&不能绑定到int常量上

const int &r2 = i;//const int&可以绑定到普通int上，那为何不考虑底层const

**constexpr和常量表达式（const expression）**

**常量表达式**是指值不会改变并且在编译过程中就能得到计算结果的表达式。

将变量声明为constexpr类型作为常量表达式的判断，而且必须用常量表达式初始化。

可以声明为constexpr的类型有：算术类型、引用和指针，被称为“字面值类型”。

constexpr指针的初始值必须是nullptr或者0，或者是存储于某个固定地址中的对象。且用限定符constexpr定义的指针只对指针有效，与指针所指的对象无关，即为顶层const。

定义抽象数据类型：

类的基本思想是抽象数据类型和封装。

数据抽象是一种依赖于接口和实现分离的编程技术。

类的接口包括用户所能执行的操作，类的实现包括类的数据成员，实现接口的函数和定义类的各种私有函数。

抽象数据类型默认定义的数据成员是私有的。而结构体类默认的数据成员是公有的，所以结构体类不是抽象数据类型。

抽象数据类型定义：成员函数的声明必须在类的内部，定义既可在类的内部也可在类的外部。定义在类内部的函数是隐式的inline函数。

例：std：：string S\_data：：isbn() const {return booNo;}

默认情况下this是指向类类型的常量指针，不允许该表this保存的地址.

等价于std::string S\_data：：isbn() const {return this->booNo;}

如上，const 的作用是修改隐式this指针的类型。this的类型为S\_data \*const，不能绑定到一个常量对象上。

常量成员函数：不能改变调用它的对象的内容。

例：伪代码，this不能显示定义

std：：string S\_data：：isbn() （const S\_data\* const this）

{return this-> booNo;}

常量对象、常量对象的引用或常量对象的指针都只能调用常量成员函数。