

### 1 比较值传递和引用传递的相同点与不同点。

值传递是指当发生函数调用时，给形参分配内存空间，并用实参来初始化形参（直接将实参的值传递给形参）。这一过程是参数值的单向传递过程，一旦形参获得了值便与实参脱离关系，此后无论形参发生了怎样的改变，都不会影响到实参。

引用传递将引用作为形参，在执行主调函数中的调用语句时，系统自动用实参来初始化形参。这样形参就成为实参的一个别名，对形参的任何操作也就直接作用于实参。

### 2 什么叫内联函数？它有哪些特点？

定义时使用关键字 inline 的函数叫做内联函数；编译器在编译时在调用处用函数体进行替换，节省了参数传递、控制转移等开销；内联函数体内不能有循环语句和 switch 语句；内联函数的定义必须出现在内联函数第一次被调用之前；对内联函数不能进行异常接口声明；

### 3 什么叫复制构造函数？复制构造函数何时被调用？

复制构造函数是一种特殊的构造函数，具有一般构造函数的所有特性，其形参是本类的对象的引用。其作用是使用一个已经存在的对象（由复制构造函数的参数指定），去初始化同类的一个新对象。

- (1) 当用类的一个对象去初始化该类的另一个对象时；
- (2) 如果函数的形参是类的对象，调用函数时进行形参和实参结合时；
- (3) 如果函数的返回值是类的对象，函数执行完成返回调用者时。

### 4 复制构造函数与赋值运算符（=）有何不同？

从概念上区分：复制构造函数是构造函数，而赋值操作符属于操作符重载范畴，它通常是类的成员函数

从原型上来区分：复制构造函数原型 `ClassType(const ClassType &);` 无返回值；赋值操作符原型 `ClassType& operator=(const ClassType &);` 返回值为 `ClassType` 的引用，便于连续赋值操作

从使用的场合来区分：复制构造函数用于产生对象，它用于以下几个地方：函数参数为类的值类型时、函数返回值为类类型时以及初始化语句

### 5 什么叫作用域？有哪几种类型的作用域？

作用域讨论的是标识符的有效范围，作用域是一个标识符在程序正文中有效的区域。

C++ 的作用域分为：函数原型作用域、函数作用域、文件作用域（全局作用域）、块作用域（局部作用域）、类作用域和名字空间作用域。

### 6 什么叫做可见性？可见性的一般规则是什么？

可见性是标识符是否可以引用的问题；

可见性的一般规则是：标识符要声明在前，引用在后，在同一作用域中，不能声明同名的标识符。对于在不同的作用域声明的标识符，遵循的原则是：若有两个或多个具有包含关系的作用域，外层声明的标识符如果在内层没有声明同名标识符时仍可见，如果内层声明了同名标识符则外层标识符不可见。

### 7 什么叫静态数据成员？他有何特点？

类的静态数据成员是类的数据成员的一种特例，采用 static 关键字来声明。对于类的普

通数据成员，每一个类的对象都拥有一个拷贝，就是说每一个对象的同名数据成员可以分别存储不同的值，这也是保证对象拥有区别于其他对象的特征的需要，但是静态数据成员，每个类只要一个拷贝，由所有该类的对象共同维护和使用，这个共同维护，使用也就实现了同一类的不同对象之间的数据共享。

#### 8 什么叫静态函数成员？他有何特点？

使用 static 关键字声明的函数成员是静态的，静态函数成员属于整个类，被同一个类的所有对象共同维护，为这些所有对象共享。静态成员函数具有以下两个方面的好处，一是由于静态成员函数只能直接访问同一个类的静态数据成员，可以保证不会对该类的其余数据成员造成负面影响；二是同一个类只维护一个静态函数成员的拷贝，节约了系统的开销，提高程序的运行效率。

#### 9 什么叫友元函数？什么叫友元类？

友元函数是使用关键字 friend 关键字声明的函数，它可以访问相应类的保护成员和私有成员。友元类是使用 friend 关键字声明的类，它的所有成员函数都是相应类的友元函数。友元不可继承，没有传递性。

#### 10 在函数内部定义的普通局部变量和静态局部变量在功能上有何不同？计算机底层对这两类变量做了怎样的不同处理导致了这种差异？

局部作用域中静态变量的特点是：它并不会随着每次函数调用而产生一个副本，也不会随着函数的返回而失效，定义时未指定初值的基本类型静态变量，会被以 0 值初始化；局部作用域的全部变量诞生于声明点，结束于声明所在的块执行完毕之时，并且不指定初值，意味初值不确定。普通局部变量存放于栈区，超出作用域后，变量被撤销，其所占的内存也被回收；静态局部变量存放于静态数据储存区，全局可见，但是作用域是局部作用域，超出作用域后变量仍然存在。

#### 11 引用和指针有何区别？何时只能使用指针而不能使用引用？

引用是一个别名，不能为 NULL 值，不能被重新分配；

指针是一个存放地址的变量。

当需要对变量重新赋以另外的地址或赋值为 NULL 时只能使用指针。引用是一个变量的别名，本身不单独分配自己的内存空间，而指针有自己的内存空间。

#### 12 比较类的 3 种继承方式之间的差别？

公有继承，使得基类 public(公有)和 protected(保护)成员的访问属性在派生类中不变，而基类 private(私有)成员不可访问。

私有继承，使得基类 public(公有)和 protected(保护)成员都以 private(私有)成员身份出现在派生类中，而基类 private(私有)成员不可访问。

保护继承中，基类 public(公有)和 protected(保护)成员都以 protected(保护)成员身份出现在派生类中，而基类 private(私有)成员不可访问。

#### 13 什么叫虚基类？有何作用？

当某类的部分或全部直接基类是从另一个基类派生而来，这些直接基类中，从上一级基类继承来的成员就拥有相同的名称，派生类的对象的这些同名成员在内存中同时拥有多个拷贝，我们可以使用作用域分辨符来唯一标识并分别访问它们。我们也可以将直接基类

的共同基类设置为虚基类，这时从不同的路径继承过来的该类成员在内存中只拥有一个拷贝，这样就解决了同名成员的唯一标识问题。

虚基类的声明是在派生类的定义过程，其语法格式为：

class 派生类名：virtual 继承方式 基类名

上述语句声明基类为派生类的虚基类，在多继承情况下，虚基类关键字的作用范围和继承方式关键字相同，只对紧跟其后的基类起作用。声明了虚基类之后，虚基类的成员在进一步派生过程中，和派生类一起维护一个内存数据拷贝。

虚基类就是为了解决多继承产生的二义性问题

#### 14 组合与继承有什么共同点和差异？通过组合生成的类与被组合的类之间的逻辑关系是什么？继承呢？

组合和继承它们都使得已有对象成为新对象的一部分，从而达到代码复用的目的。组合和继承其实反映了两种不同的对象关系。

组合反映的是“有一个”（has-s）的关系，如果类 B 中存在一个类 A 的内嵌对象，表示的是每一个 B 类型的对象都“有一个”A 类型的对象，A 类型的对象与 B 类型的对象是部分整体的关系。

继承反映的是“是一个”（is-a）的关系，在“是一个”关系中，如果类 A 是类 B 的公有基类，那么这表示每一个 B 类型的对象都“是一个”A 类型的对象，B 类型的对象与 A 类型的对象是特殊与一般的关系。

#### 15 基类与派生类的对象，指针或引用之间，哪些情况下可以隐含转换，哪些情况下可以显示转换？在涉及多重继承或虚继承的情况下，在转换时会面临哪些新问题？

派生类指针可以隐含转换为基类指针，而基类指针要想转换为派生类指针，则转换一定要显示地进行。因为从特殊的指针转换到一般的指针时安全的，因此允许隐含转换；从一般的指针转换到特殊的指针是不安全的，因此只能显示地转换。基类对象一般无法被显式转换为派生类对象。在多重继承情况下，执行基类指针到派生类指针的显式转换时，有时需要将指针所存储的地址值进行调整后才能得到新指针的值。但是，如果 A 类型是 B 类型的虚拟基类，虽然 B 类型的指针可以隐含转换为 A 类型，但是 A 类型指针却无法通过 static\_cast 隐含转换为 B 类型的指针。

#### 16 什么叫做多态性？在 C++ 种是如何实现多态的？

多态是指同样的消息被不同类型的对象接收时导致完全不同的行为，是对类的特定成员函数的再抽象。C++ 支持重载多态，强制多态，包含多态和参数多态。

在基类中声明相应的函数为 virtual 型，然后在派生类中实现该函数，这样就可以通过基类指针调用派生类对象的函数，实现了运行时动态绑定，即多态的功能。

#### 17 什么叫抽象类？抽象类有何作用？抽象类的派生类是否一定要给出纯虚函数的实现？

带有纯虚函数的类是抽象类。其主要作用是通过它为一个类族建立一个公共的接口，使他们能够更有效地发挥多态特性。抽象类声明了一个类族派生类的共同接口，而接口的完整实现，即纯虚函数的函数体，要由派生类自己定义。抽象类派生出新的类之后，如果派生类给出所有纯虚函数的函数实现，这个派生类就可以定义自己的对象，因而不是抽象类；反之，如果派生类没有给出全部纯虚函数的实现，此时的派生类仍然是一个抽象类。

18 在 C++ 中，能否声明虚构造函数？为什么？能否声明虚析构造函数？有何用途？

在 C++ 中，不能声明虚构造函数，多态是不同的对象对同一消息有不同的行为特性，虚函数作为运行过程中多态的基础，主要是针对对象的，而构造函数是在对象产生之前运行的，因此虚构造函数是没有意义的；

可以声明虚析构造函数，析构造函数的功能是在该类对象消亡之前进行一些必要的清理工作，如果一个类的析构造函数是虚函数，那么，由它派生而来的所有子类的析构造函数也是虚函数。析构造函数设置为虚函数之后，在使用指针引用时可以动态联编，实现运行时的多态，保证使用基类的指针就能够调用适当的析构造函数针对不同的对象进行清理工作。

19 什么叫做流？流的提取和插入是指什么？I/O 流在 C++ 中起着怎样的作用？

流是一种抽象，它负责在数据的生产者和数据的消费者之间建立联系，并管理数据的流动，一般意义下的读操作在流数据抽象中被称为（从流中）提取，写操作被称为（向流中）插入。操作系统是将键盘、屏幕、打印机和通信端口作为扩充文件来处理的，I/O 流类就是用来与这些扩充文件进行交互，实现数据的输入与输出。

20 什么叫做异常？什么叫做异常处理？

当一个函数在执行的过程中出现了一些不平常的情况，或运行结果无法定义的情况，使得操作不得不被中断时，我们说出现了异常。异常通常是用 throw 关键字产生的一个对象，用来表明出现了一些意外的情况。我们在设计程序时，就要充分考虑到各种意外情况，并给与恰当的处理。这就是我们所说的异常处理。

21 C++ 的异常处理机制有何优点？

C++ 的异常处理机制使得异常的引发和处理不必在同一函数中，这样底层的函数可以着重解决具体问题，而不必过多地考虑对异常的处理。上层调用者可以在适当的位置设计对不同类型异常的处理。

22 比较函数重载和虚函数在概念和使用方式方面有什么区别？

- 1) 函数重载可以用于普通函数（非成员的函数）和类的成员函数，而虚函数只能用于类的成员函数。
- 2) 函数重载可以用于构造函数，而虚函数不能用于构造函数。
- 3) 如果对成员函数进行重载，则重载的函数与被重载的函数应当都是同一类中的成员函数，不能分属于两个不同继承层次的类。函数重载是横向的重载。虚函数是对同一类族中的基类和派生类的同名函数的处理，即允许在派生类中对基类的成员函数重新定义。虚函数的作用是处理纵向的同名函数。
- 4) 重载的函数必须具有相同的函数名，但函数的参数个数和参数类型二者中至少有一样不同，否则在编译时无法区分它们。而虚函数则要求再同一类族中的所有虚函数不仅函数名相同，而且要求函数类型、函数的参数个数和参数类型都全部相同，否则就不是重定义了。也就不是虚函数了。
- 5) 函数重载是在程序编译阶段确定操作的对象的，属静态关联。虚函数是在程序运行阶段确定操作的对象的，属动态关联。

23 比较 class 和 struct 结构体的异同

不同之处：

- 1) 默认成员访问权限不同，struct 默认是 public，而 class 默认是 private。

- 2) 默认继承方式不同, struct 默认是 public, 而 class 默认是 private。
  - 3) 在定义模板参数时可以用 class 但不能用 struct。
- 相同之处在于: 都可以包含成员函数, 都可以实现继承, 都可以实现多态。

## 24 浅拷贝和深拷贝的区别?

在某些情况下, 类内的成员变量需要动态开辟堆内存, 如果实行按位拷贝, 就可能会出现两个对象里面的成员变量指针指向同一块内存区域。当回收其中一个对象时, 该对象成员变量所指向的内存区域也将回收, 会导致另一个对象存在野指针的情况。非常危险。

深拷贝和浅拷贝可以简单理解为: 如果一个类拥有资源, 当这个类的对象发生复制过程的时候, 资源重新分配, 这个过程就是深拷贝, 反之, 没有重新分配资源, 就是浅拷贝。

## 25 const 和 static 作用

const 作用:

- 1) 定义 const 常量, 具有不可变性, 保护被修饰的东西。防止意外修改, 增强程序的健壮性。
- 2) 便于类型检查, 使编译器对处理内容有更多的了解。
- 3) 可以避免意义模糊的数字出现, 也便于进行参数的修改。同宏定义一样, 可以做到不变则以, 一变都变。
- 4) const 定义的常量在程序运行过程中只有一份拷贝, 而宏定义却又若干份拷贝。

static:

- 1) 作为类的静态数据成员: 被类的所有对象共同拥有, 在内存中只有一份拷贝。在类内声明, 在类外初始化。
- 2) 在全局变量前加上 static 关键字, 该变量被定义为一个全局变量, 特点:
  - a) 变量在全局数据区分配内存
  - b) 如果不显示初始化, 则隐式的初始化为 0
  - c) 作用域仅限于本文件内
- 3) 在局部变量前加 static 关键字, 该变量被定义为静态局部变量, 特点:
  - a) 在全局数据区分配内存
  - b) 如果不显示初始化, 则隐式的初始化为 0
  - c) 始终驻留在全局数据区, 直到程序结束。作用域仅限于函数体内, 当函数执行完后作用域随之消失。
- 4) 静态函数, 特点:
  - a) 静态函数只能在本源文件中使用
  - b) 在文件作用域下声明的 inline 函数默认是 static 的

## 26 析构函数带 virtual 和 不带 virtual 的区别

如果基类要被派生, 则析构函数一定要带 virtual。否则, 通过基类指针指向派生类对象后, 删除该指针时并不会释放派生类对象的空间。也即, 不会调用派生类的析构函数。

不带 virtual 的几种情况:

- 1) 作为非公有基类
- 2) 不作为接口使用的基类
- 3) 可以保证这个类不被 public 继承
- 4) 如果他的所有派生类的 virtual 函数都是什么都不做
- 5) 如果不需要基类指针指向派生类对象



## 27 重写，重载和隐藏的区别？

重载：在一个类中的同名函数，但参数列表不同，函数的返回值相同。

隐藏：派生类的函数屏蔽了与其同名的父类函数

重写（覆盖）：父类和子类存在相同的函数，包括返回值，参数列表均相同，父类的该函数必须含有 virtual 关键字。

## 28 C++中的四个默认函数

构造函数：先基类后派生类

拷贝函数：对象被创建时调用

调用拷贝构造函数的情况：

- a) 一个对象以值传递的方式传入函数体
- b) 一个对象以值传递的方式从函数返回
- c) 一个对象需要通过另外一个对象初始化

析构函数：每个类只有一个析构函数，

赋值函数：

## 29 不可以被重载的运算符--五个

除了类属关系运算符“.”、成员指针运算符“.\*”、作用域运算符“::”、sizeof 运算符和三目运算符“?:”

## 30 三种字符数组初始化的方法

```
char str[5] = {'a', 'b', 'c', 'd', 'e'};
```

```
char str[6] = "abcde";
```

```
char str[] = "abcde";
```

## 31 哪些操作符必须重载为成员函数

=    ->    ()    []    new    delete

## 32 什么是“else 摇摆问题”，举例说明

C++编译器总是把 else 同与它最近的 if 联系起来，因此如果没有正确放置花括号就会出现逻辑错误。

## 33 函数模板和函数重载的区别与联系

**区别：**

函数重载：C++允许使用同一个函数名，不同的参数个数和参数类型来定义多个函数。重载函数的参数个数，类型，顺序至少有一个不同。返回值可以相同，也可以不同。

函数模板：是对一类同构函数的抽象定义，并不是具体的函数，函数模板的定义被编译时不会产生任何可执行的代码。

**联系：**

针对不同类型的数据，但实现相同功能的函数重载可以通过函数模板来替代。

## 34 怎样区别虚函数和纯虚函数？两者都有什么作用

纯虚函数是在基类中声明的虚函数，在基类中没有定义。在基类中实现纯虚函数的方法是在函数原型后加“=0”。虚函数可以实现也可以不实现。

定义纯虚函数是为了实现一个接口，起到了一个规范的作用。规范继承这个类的程序员必须实现这个接口。声明了纯虚函数的类为抽象类，不能实例化对象。

定义虚函数是为了允许用基类的指针来调用子类的这个函数。

**抽象类的作用**：将有关的操作作为一个接口组织在一个继承层次结构中，由他来为派生类提供一个公共的根。派生类将具体实现在其基类中作为接口的操作。

35 面向对象程序“接口与实现方法分离”，有什么作用（好处）

接口与实现分离技术可以只把接口暴露给用户，具体的实现细节隐藏起来，当需要改动代码时，只要在实现部分修改后编译，用户无需再编译自己的项目。

36 列出所有与字符串处理有关的头文件

string.h 是 C 语言中字符串操作函数的头文件

cstring 是 c++ 对 C 语言中的 strcpy 之类的函数申明，包含 cstring 之后，就可以在程序中使用 C 语言风格的 strcpy 之类的函数。

string 是 c++ 语言中 string 类模板的申明

CString 是 MFC 中定义的字符串类，MFC 中很多类及函数都是以 CString 为参数的

37 C++ 中显示类型转换

- 1) static\_cast：最常用的类型转换，提供基本数据类型之间的转换操作。
- 2) const\_cast：可以把 const 类型的指针转换为非 const 类型的指针。
- 3) reinterpret\_cast：解释的意思，主要用于不相关类型之间的转换。目标和原始值之间至少有相同的位数，我们可以将转换之后的值再转换回去，而不像其它 3 种类型可能会导致精度丢失
- 4) dynamic\_cast：运行时检查该转换是否类型安全，但只有在多态类型时合法，主要用于类层次间的上行和下行转换，还可以用于类之间的交叉转换。在进行类层次间的上行转换时和 static\_cast 效果一样，进行下行转换时，具有类型检查的功能，比 static\_cast 更安全。

38 delete 和 delete [] 的区别？

delete ptr 代表用来释放内存，且只用来释放 ptr 指向的内存。

delete[] rg 用来释放 rg 指向的内存！！还逐一调用数组中每个对象的析构函数

对于像 int/char/long/int\*/struct 等等简单数据类型，由于对象没有 destructor，所以用 delete 和 delete [] 是一样的！但是如果是 C++ 对象数组就不同了！

39 存储类别说明符可划分为两类。**静态存储类别**（static，extern）和**自动存储类别**（auto，register）。这样的变量在程序执行进入到定义它们的程序块时创建，在程序块激活时存在，在程序块退出时销毁。

40 什么是运算符重载，它如何增强 C++ 的扩展性？

运算符重载是对已有的运算符赋予多重含义，使得同一个运算符作用于不同类型的数据时导致不同的行为。

运算符重载的本质是函数重载，可以改变现有运算符的操作方式，以适用于类类型，为类的用户提供了一个直接的接口，使得用户程序所用的语言是面向问题的，而不是面向机器

的，增强了 C++ 的扩展性。

41 为什么说“继承是 C++ 面向对象的主要特征之一”？请简要说明。

继承是一种联结类的层次模型，层次结构的上层是最具有通用性的，而下层的部分，即后代具有特殊性。类可以从他的祖先那里继承方法和成员变量，也可以增加新的方法是适用于特殊的需要。如果没有继承，类就缺失了一块层次结构，代码重用和数据共享就贯彻不到底，有了继承，就会有抽象编程中的多态问题，只有从机制内部真正解决了多态表现问题，对象的数据封装，信息隐藏，代码重用等招式才能淋漓尽致的发挥出来。才称得上真正的面向对象编程。

42 作用域辨析？

**类作用域：**

在类的作用域内，类的成员可以被类的所有成员函数直接访问，在类的作用域之外，public 类成员通过对象的句柄之一而引用。句柄可以是对象名称，对象的引用，对象的指针。对象，引用或指针指定了客户可访问的接口。

**全局名字空间作用域：**

声明与任何函数或者类之外的标识符具有全局名字空间作用域，这种标识符对于从其声明处开始直到文件结尾处为止出现的所有函数都是已知的，即可访问的。位于函数之外的全局变量。函数定义和函数原型都属于该作用域。

**函数作用域：**

标签是唯一具有函数作用域的标识符，标签可以在函数中的任何地方使用，但是不能在函数体外被引用。

**局部作用域：**

在一个语句块中声明的作用域，开始于标识符的声明处，结束于标识符声明处所在语句块的结束右花括号处。局部变量具有局部作用域，函数形参也是函数的局部作用域。当语句块是嵌套的，并且外层语句块的一个标识符和内层语句块的一个标识符具有相同的名字时，外层的标识符处于隐藏状态，直到内存的语句块执行结束为止。

**函数原型作用域：**

函数原型形参列表中的标识符，在一个函数原型中，一个特定的标识符只能使用一次。