2014年期末C++复习资料

——计科134班出品

使用前必读：

1. **本资料仅供参考，复习时必须与教材同时使用。**
2. **复习时应当对照本资料上的内容复习教材上的难点和例题，切不可先看教材再看本资料或者先看本资料再看教材。**
3. **本资料中内容如有与任课教师所讲内容冲突的，以任课教师为主。**
4. **计科134出品，必属精品。转载请保留此行。**

# 第一章

本章为C++的一个初步介绍，几乎所有内容在后面的章节中都有详细的讲解，故考试时很少涉及第一章内容。同学们在复习本章时只需掌握基本概念即可，注意下面几点：

**1.3.4 函数的重载**，掌握概念（**凡本文涉及的知识点都要求掌握概念，下面不在特殊说明**），**例1.6 例1.7** 。要注意的是：重载函数与原函数相比，**要么参数个数发生改变，要么参数类型发生改变**。例如：void function( int x , int y )（原函数）与void function( int x , int y , int z )或void function( int x , float y )是重载函数。（函数重载在我们目前学习的c++中最大体现就在于构造函数的重载。）

**1.3.6 有默认参数的函数 重点** 第三章中“带默认参数的构造函数”是必考内容，这里要牢牢掌握什么是默认参数以及默认参数的用法。注意：默认参数一定在参数列表的最右端如：void f1（flaot a，int c，int b=0）

**1.3.7 变量的引用** 主要看“将引用作为函数参数” **例1.12** “对引用的进一步说明”应做了解。引用的作用就是替代指针，并且不会产生地址方面的问题。

**1.3.9 作用域运算符** 掌握“：：”运算符的意义，该运算符在后面大量出现。例如void Student :: display( )表示display函数是Student类的成员函数，其类型为void。此处注意语句顺序为：类型、类名、作用域运算符：：、函数名/变量名

**1.3.10 字符串变量 重点** 熟练掌握本节所有内容。在程序中使用字符串变量时注意添加头文件#include<string>。使用时将string型变量当做一般变量即可，例如：定义整型变量a写作int a ，定义字符串变量a就应写作string a ，但是不同于其他类型，string不会变蓝。 **例1.17**

**1.3.11 动态分配/撤销内存运算符 new和delete 考点** 平常最考察的就是构造函数用指针做形参时需要开辟内存，否则会因为不明了指针指向的地址块大小而报错，如在我们做过的TV类中：

TV(char \*m,int s,int u)

{

model=new char[strlen(m)+1];

strcpy(model,m);

}

注意：只要有new开辟空间一定要有delete删除空间，即有多少个new就要有多少个delete。

# 第二章

本章主要介绍C++中的类和对象，两者是C++的灵魂，应牢牢掌握

**2.2.1 类和对象的关系 基础** 对象是类的衍生，类的具体，类就像模具，对象就像是模具产出的产品，只不过每个产品编码不同。

**2.2.2 声明类类型** 掌握**P45下方的程序段**。习惯上在定义类时先写public成员再写protected成员再写private成员（并非规定，习惯而已）。

**2.2.3 定义对象的方法 重点** 掌握第一种方法，了解第二种方法，无视第三种方法。（手里没有教材的话你肯定不知道我在说什么-\_-！！）

**2.3 类的成员函数 重点** 掌握50页上方的程序段，熟练运用void Student :: display( )这种定义方法（又称**类外定义**）。注意顺序：类型、类名、作用于运算符、函数名。2.3.3和2.3.4不看。

**2.4 对象成员的引用 重点 其中三个方法都要熟练运用** 55页上方“在类外只能…….是错误的”是很多人易犯的错误。2.4.2的内容在今后的学习中很常见，即使有了引用，指针依然非常具有存在感（指针是很喜闻乐见的……）

**2.5 类和对象的简单应用举例 重点** 本节只看两个例题：**60页例2.3和61页例2.4** 。 务必掌握每一个细节，尤其是例2.4。

**2.6 类的封装性和信息隐蔽 无视**

**本章课后习题要求会做1、2、3题**

# 第三章（重点）

本章讨论构造函数等C++中的重要工具，更是考试中最喜闻乐见的章节。本章知识点多且复杂，对于C++的初学者来说是有一定难度的。全书本章的难度个人认为排行第三，复习时应多在本章分配时间。

同学们无论在复习、考试还是自己编写程序是要注意，自己设定类名、函数名、变量名等名称时，要使用含义明确、简洁明了的词汇。例如学生类就是Student（拼音亦可），这样可以大大增强程序的可读性。考试遇到程序阅读题时，如果题目中的名称太长会影响审题，可以自己改成a、b、c等习惯的名字，防止出错。

**3.1 构造函数 重点中的重点，不仅是本章，全书中凡是提到构造函数的都是重点** 首先无视3.1.1 。掌握**例3.1**，注意：**1.构造函数名必须与类名相同。2.构造函数用于对数据成员进行初始化。3.构造函数没有类型也没有返回值。4.构造函数可以有参数也可以没有参数。根据题目要求写构造函数是填空和大题的必考知识点**。掌握**71页下方在类外定义构造函数**的方法。**例3.2** 。3.1.4内容虽少但是非常重要

**Box::Box( int h, int w, int len):height(h),width(w),length(len){ }**

**等效于**

**Box::Box( int h, int w, int len){ height=h; width=w; length=len; }**

前者被称为**参数初始化表**，用参数初始化表初始化数据成员必考。**例3.3** 。关于构造函数的重载只需按照第一章函数重载的概念和注意事项即可。注意区分以下几种构造函数：

**Box::Box( int h, int w, int len){ height=h; width=w; length=len; }带参数的构造函数**

**Box::Box(int h=0, int w=0, int len=0){ height=h; width=w; length=len; }带默认参数的构造函数**

**Box::Box( ){ height=0; width=0; length=0; } 默认构造函数**

可以发现带默认参数的构造函数兼顾了其他两种构造函数的功能，在实际应用中这类构造函数应用得最多。注意带默认参数的构造函数也属于默认构造函数，这样的构造函数在一个类中只能有一个。**例3.4** 。

**3.2 析构函数** 如果要考析构函数必考析构函数的执行顺序。在了解析构函数之前应先了解什么叫对象的释放。释放就是清空该对象所占的内存空间，可以简单地理解为删除当对象被释放时，会执行析构函数。**牢记78页中部执行析构函数的4个条件**。同学们要注意一个大误区，**析构函数不是释放对象的函数，无论程序是否有析构函数，对象都要释放。析构函数的作用是在对象释放时做一些额外的工作**，例如：cout<<“我被释放啦！”<<endl；在答程序设计题时，如题目没有要求写析构函数就不要写，万不可盲目跟风。

**3.3 调用构造函数和析构函数的顺序** 可能会出一个选择题，如果复习压力较大可以无视。

**3.4 对象数组** 此物是结构体数组的升级版，用法与结构体数组基本相同。**例3.6**

**3.5 对象指针** 如果同学们能抽出半小时时间回顾一下上学期程序设计基础中关于指针的内容，复习起来会容易得多。**掌握3.5.1和3.5.2的所有相关概念**，无视3.5.3

**3.6 共用数据保护** 本节的知识点难度不高但极易混淆。应**先掌握概念部分**，再**牢记90页至96页的表3.1 表3.2 表3.3** 。注意区分const的位置不同带来的变化，例如：const Time \*p是**指向常对象的指针**p，Time const \*p 是**指向对象的常指针**p。这些内容靠死记硬背很难记住，最佳的方法是上机实验。**上机实验是学习程序设计语言的万能钥匙，可以解决学习中的一切难题。**3.6.5 对象的常引用请结合前面对象的引用来复习，**例3.8** 。

**3.7 对象的动态建立与释放 无视（但是本节知识在大二上学期学习链表时是重点）**

**3.8 对象的赋值与复制 例3.9** 。本节又向我们介绍了一种新的构造函数——拷贝构造函数。**重点掌握99页下方的概念和程序段**，拷贝构造函数是常考点（多出现在填空题）

**3.9 静态成员** 掌握一下静态成员的作用即可。注意一点：静态成员的初始化只能在类外进行。

**3.10 友元** 友元函数在运算符重载（第四章）中有重要应用。本节重点掌握友元的概念、作用和用法。完全掌握**例3.12 例3.13** 。注意定义友元成员函数之前要先声明互为友元的两个类。

**3.11 类模板 无视（但是本节知识在大二上学期很常用）**

**本章习题要求会做2、3、5、6、7、8、10**

# 第四章（非重点）

本章节考的内容很少，但是知识点层次复杂难懂，以考试为目标的同学还是要加以理解。弄清内部逻辑。

**4.1 什么是运算符重载 例4.1** 注意本例中的complex\_add函数，该函数即重载函数的原型。它有一个形参，函数体中的**c.real=real+c2.real语句中的real实际上是c1.real**。原因为该函数是成员函数，主函数中有调用语句c1.complex\_add( )。若该函数是友元函数，它的形参应写成**（Complex& c1 , Complex& c2）**，c.real=real+c2.real语句也应改成**c.real=c1.real+c2.real**。这一点非常重要，它体现了将友元函数作为重载函数的特点，即**友元函数做重载函数重载双目运算符时运算符两侧的变量可以交换位置**。如果重载“+”运算符用于复数的加运算是，如果想满足加法分配率，就要使用友元函数。

**4.2 运算符重载的方法 掌握例4.2**熟记**函数类型+operator+运算符+（形参表列）**的模式。掌握例4.2后应具备独立编写对复数减法、乘法和除法的重载的程序的能力。

**4.3 重载运算符的规则** 记住125页关于重载运算符的几个规则，记住不能重载的5个运算符。

**4.4 运算符重载函数作为类成员函数和友元函数** 本节详细介绍了4.1中所说的友元函数的特点的问题。**掌握例4.3以及书上对例4.3的分析**

**4.5和4.6可以无视**

**4.7 重载流插入和流提取运算符** 本节只要求同学们认识还有这样一种重载函数即可，如果在程序阅读题中遇到要能理解它的作用。对于想把专业课学好的同学应先掌握第7章的内容在来看本节内容。

**4.8 不同类型数据间的转换** 4.8.1无视。4.8.2和4.8.3在复习时合为一体，这两节之间的关系极其紧密。这里我们又认识了一种新的构造函数——**转换构造函数**。务必掌握它的概念和用法。**例4.9 例4.10** 。

**本章习题要求会做2、3、5、6**

# 第五章（重点，必出大题）

第五章和第六章荣幸地并列成为本书最重要，最困难，最变态的章节，同时也是考试中占据大分值的章节。考试中无论是小题还是大题均能看到这两章的身影。所以，对这两章的掌握是及格和高分的关键。**只希望及格就好的同学应当避难就易，先巩固自己会的，再争取拿下一些比较难的，实在看不懂的应当放弃。考试时重心放在考查基本概念的题目，大题想到什么写什么，切忌少写或不写；想拿奖学金的同学要把本资料的每一个难点和例题都掌握；想把专业课学好的同学最为简单暴力，请无视本资料，直接把全书融会贯通即可。**

**5.1 继承与派生的概念** 纯概念章节，对本章内容理解得比较好的同学可无视；理解得不太好的需死记。

**5.2 派生的声明方式** 注意：若想让B类成为A类的派生类，应在声明**B类**时添加class B public（也可以是其他继承方式）A。同时A类**必须已经存在**。即在声明派生类时指定它的基类。

**5.3 派生类的构成** 又是纯概念章节。注意：**虽然派生类不能访问基类的私有成员，但是基类的所有成员都通过继承给了它的派生类**。之所以设定派生类不能访问基类的私有成员，是为了维护类的封装性，增强信息安全。这里复习一下C++的三大特性：**封装性、继承性、多态性**。

**5.4 派生类成员的访问属性** 本章核心考点。首先列举三种继承方式。

表4.1 基类成员在派生类中的访问属性

|  |  |  |
| --- | --- | --- |
| 基类中的属性 | 继 承 方 式 | 在派生类中的访问方式 |
| 公有成员 | 公有继承 | 公有 |
| 保护成员 | 保护 |
| 私有成员 | 不可直接访问 |
| 公有成员 | 保护继承 | 保护 |
| 保护成员 | 保护 |
| 私有成员 | 不可直接访问 |
| 公有成员 | 私有继承 | 私有 |
| 保护成员 | 私有 |
| 私有成员 | 不可直接访问 |

总结：**在任何情况下派生类均不能访问基类的私有成员，只有通过派生类的类对象调用基类公有成员函数来访问基类的私有成员。**

**156页至162页表5.1 表5.2 表5.3 表5.4 本节所有例题（尤其是例5.1）。**考试重点考查的是公有继承，其他两种继承方式不太常见。多级派生时要综合上述三种继承来分析，**掌握165页下方用列表的形式分析各成员在各级派生类中是何种成员的方法**，在考试时可以很快地理清题目中程序的访问属性。

**5.5 派生类的构造函数和析构函数** **例5.5 注意派生类的构造函数在声明时仅列出从它的上一级（直接基类）继承的成员，不需要从第一级开始一级一级地列出。**例如：Student1(int n , string name , char s , int a , string ad) : Student(n , name , s){age=a ; addr=ad ;}中Student(n , name , s)**实质上是一个函数调用语句，调用基类构造函数为派生类中从基类继承来的数据成员赋初值**。Student(n , name , s)里的三个参数是实参，{age=a ; addr=ad ;}该语句为派生类特有的数据成员赋值。

**子对象**是在派生类的私有成员中定义**基类的**对象，又被称为“**类类型数据成员**”。**例5.6**详细地说明了有子对象派生类的构造函数的写法和子对象的作用，应重点掌握。**例5.7** 。派生类的析构函数了解即可。

**5.6 多重继承** **例5.8** 多重继承的难点在于二义性问题，**179页图5.14**和**180页图5.15**详细描述了这个问题。解决该问题有两种方法：1.在成员前加类名和作用域运算符，不推荐。**2.虚基类。**掌握虚基类的概念和作用，注意下面的代码：

**class A**

**class B: virtual public A**

**class C: virtual public A**

**class D: public B, public C**

B类和C类均继承于A类，D类同时继承于B、C两类。但是由于使用了虚基类，A类的成员在D类中只保留一份。**例5.9**

**5.7至5.9看一下概念即可，其他无视。对于5.8有一点特殊说明：如果考试中程序设计题要求使用类的组合，请无视这条要求，直接使用继承或者友元类即可。**

**本章习题要求会做1、2、3、4、5、6**

# 第六章（重点，必考大题）

第六章主要研究多态性及虚函数，本章的重要性请参考上一章的介绍。

**6.1 多态性的概念** 本节为6.2做准备，看一下概念。多态性的核心思想是：想不同的对象发送同一个消息，不同的对象在接收时会产生不同的行为。

**6.2 一个典型的例子** 本节的名称很好地体现了它的内容，例6.1是本书中最长、最复杂、最变态的例题（没有之一），它还综合了第三、四、五章的知识。为了让同学们能更好地掌握本例，建议先对本例作如下改动：

**1.将例题中所有的“get”改成“show”**。“get”容易让人联想起处理字符的函数getchar( )从而引起混淆，实际上本例中“get”仅仅是个函数名而已。

**2.去掉所有的“const”**。本例中“const”的目的是帮助大家回忆第三章中常成员函数的知识，const的作用是使该函数只能引用数据成员而不能改动数据成员。由于考试中对它的考查不多，所以不宜出在在这里干扰视线。

同学们对本例以及书中对本例的分析要做到“牢记一切，字字理解”。下面列举几个本人认为在复习本例中容易混淆的地方：

**1.**本例中包含3个类，他们分别是“Point”、“Circle”、“Cylinder”。在考试中如果单词记不住了没有关系，我们可以这样定义：“Dian”、“Yuan”、“Yuanzhu”。要知道名字只是一个代号……

**2.**函数float getX ( ) const { return X ; }经过上述改动后变为float showX ( ) const { return X ; }。**它实际上等效于void display ( ) { cout<<X<<endl; }**（这是我们在前面的章节中经常看到的形式，理解了这一点就不会看不懂该函数）

**3.**要求**一眼看出**friend ostream& operator << ( ostream& , Circle& )（及其他类似函数）是重载流插入运算符函数作友元函数，作用是使运算符“<<”可以直接输出类对象。如果不能做到这一点要立即回顾第四章4.7的内容。**本例对多态性的体现就在这个函数中**，**重点理解204页顶部“在本例中存在静态多态性…….”这段话**。

**4.只有203页的main函数才是该程序的真正主函数**，其他的都用于测试定义的类。以圆为例：测试圆类的main函数先定义了一个圆心为（3.5 ，6.4），半径为5.2的圆，计算并输出后圆心改为（5 , 5），半径改为7.5再进行计算和输出。3个用于测试的main函数都是“定义——>计算——>输出——>改变——>计算——>输出”的模式。**在编写代码很长的程序时，往往不是等到全部写完再测试，而是每写一部分就用函数测试，这样可以降低调试程序的难度和时间。**

**6.3 虚函数（动态多态性）** 牢记概念，掌握虚函数的定义和作用。总体而言，若基类和派生类中有**类型、函数名、参数个数和参数类型都相同而函数体不同**的函数，则不能通过改变指向对象的指针的指向来调用它们。但如果**在基类中声明**该函数为虚函数，就能用一个**指向基类对象的指针**调用派生类的该函数。具体做法是先定义派生类对象，再让指针指向它，再调用函数。例如，有下面的代码：

**class A**

**{**

**public:**

**virtual void function(){cout<<"1"<<endl;}**

**};**

**class B: public A**

**{**

**public:**

**void function(){cout<<"2"<<endl;}**

**};**

**int main()**

**{**

**A a; //**定义A类对象a

**A \*p=&a; //**定义指向A类对象的指针p，并使它指向a

**p->function(); //**调用function函数

**B b; //**定义B类对象b

**p=&b; //**使p指向b

**p->function(); //**调用function函数

**}**

若在定义A类时没有加virtual，则两次调用均输出“1”，因为两次都调用了A类的function。添加virtual后，第二次调用输出“2”，因为第二次调用了B类的function。**例6.2（本例和例6.1同一重量级，你们懂的）**，**掌握206页和207页的所有内容，注意206页下方虚函数的声明方法和207页下方虚函数的4条使用方法。6.3.2无视。6.3.3要记两个使用虚函数的注意事项和将成员函数声明为虚函数时需要考虑的4点。**

**虚函数和虚基类的区别：**

1. 虚函数的作用是通过改变指向基类对象的指针的指向来访问不同派生类中类型、函数名、参数个数和参数类型都相同而函数体不同的函数。虚基类的作用是在多重继承中间接派生类只保留一份基类的成员。
2. 二者声明方式不同，详细请参考各自的章节

**6.3.4虚析构函数 无视，但同学们要知道析构函数也能被声明成为虚函数，选择题中可能出现。**

**6.4纯虚函数与抽象类 无视。**

**第六章课后习题不做。**

# 第七章和第八章

这两章在考试中**几乎不涉及**，所以我们的复习策略是：**当且仅当已经把前六章的内容复习好时，才看这两章**。前六章的内容足够考90分以上了。但是想把专业课学好的同学，你们懂的…… 本资料就不列举这两章的重要知识点和例题了。

至此本资料所有内容结束，本人能力有限，如果文中出现任何纰漏或错误请同学们及时对照教材或请教老师更正。最后祝同学们都考出好成绩！

注：本复习资料基于计科123班杜鸿舰的c++复习资料加以增删改编