《面向对象程序设计及 C++》总复习

（2024/25 第二学期）

考试形式（开卷）

一、 程序阅读：4*5’=20’

二、 程序填空：15*2’=30’（4 小题）

三、 编程题 1：15’

四、 编程题 2：15’

五、 编程题 3：20’

总评=平时×50%+期末×50%

第 1 章：面向对象程序设计及 C++语言概述

需要掌握的知识要点：

（1）**面向对象的三个主要特性：封装性、继承性、多态性**

封装性：使得面向对象的程序设计具有面向过程无法比拟的安全性和可靠性。将属性和行为作为一个整体，并对其行为和属性加以控制。**public，private，protected。**

继承性：提高代码可重用性的重要措施。**多层继承，多重继承。**

多态性：一种行为对应多种不同的实现方法。**静态多态，动态多态。**

（2）**C＋＋源程序的扩展名：**.cpp；编译后的**目标文件的扩展名：**.obj；链接后的**可执行文件的扩展名**为：.exe；程序总是从 main()函数开始运行。

（3）**对象和类的关系：**

- 一个类的所有数据成员以及成员函数是一个被封装的整体，从使用的角度看，类只是一个抽象的概念，只有生成具体的对象才有意义。
- 对象是类的一个具体的个体，也称为类的一个实例。
- 类与对象的关系类似于类型和变量的关系。

（4）**面向对象设计语言：**

Simula，Smalltalk， Objet-C， Eiffel， Ada ，C++， Java， Python。

第 2 章：C++语言对 C 语言的改进及扩展

需要掌握的知识要点：

（1）新的输入/输出及注释方式 ：I/O 流对象 cin、cout。

需要用#include <iostream>及 using namespace std；进行文件包含。

（2）名字空间的定义及使用：防止命名冲突，用户可以根据程序的需要自行定义和使用名字空间。3 种方式使用名字空间内容。

（3）新增字符串的处理：C++新增 string 类型处理字符串。

需要用#include <string>包含头文件 string。

（4）函数相关的改进：

- 使用作用域运算符（::），可扩大全局变量的作用域，能在局部变量作用域内访问同名全局变量（全局变量默认值为 0）
- 形式参数参数可带有默认值，（形参从右向左、实参从左向右）。
- 函数重载：功能相同或类似，只是在形式参数的个数、类型、顺序方面有区别的不同函数，可以用相同的函数名来命名。

（5）引用的定义与应用： 声明时通过“&”来标记，用来为变量起别名，主要用作形式参数以及作为函数的返回值，与所代表的变量等价。

（6） 动态内存空间管理：new 和 delete 申请和释放动态内存空间。

特别提醒：不能用 delete 指针变量名；释放动态一维数组空间，会导致后 N-1 个元素空间未释放。

（7） 异常处理：try（检查异常）、throw（抛出异常）、catch（捕捉异常）。

练习：

1. 作用域运算符 ::

该程序运行输出是：100,101 。

```
# include<iostream>
using namespace std;
int i = 15;
int main()
{ int i;
    i = 100;
    : : i=i+1;
    cout<<i<<”,”<< ::i<<endl;
    return 0;
}
```

（课后习题 3 四-3）

2. 引用（变量的别名）

引用的定义，引用、指针作为函数参数。

该程序运行输出是：1000 1000 。

```
# include<iostream>
using namespace std;
 void s(int *a,   int &b)
 {   int *t=a;
          *a=b;
          b=*t;
```

```
}
void main()
{    int x=500, y=1000;
     s(&x, y);
     cout<<x<<"    "<<y<<endl;
}
```

（课后习题 3 四-6）

第 3 章：类与对象的基本知识

需要掌握的知识要点：

（1） 类与对象的定义：定义格式，定义对象，访问类的数据成员，类成员函数的两种实现方式。内联函数，关键字：**inline**。

（2） 访问属性：**public**，**private**，**protected**。

（3）**this** 指针： 每个成员函数都有的特殊隐含指针，使用对象时，系统自动生成，指向当前对象，可以在程序中显式使用。

（4）构造函数与析构函数：默认构造函数（无参数），无参构造函数，具有默认参数的构造函数，复制构造函数（参数通常为对象的常引用），析构函数（一个类只有一个析构函数），析构函数与动态内存分配（如果构造函数中用 **new** 申请了动态内存空间，当生命周期结束时，通常定义一个析构函数释放所有申请的动态空间）。

（5）浅复制与深复制：浅复制会造成指针悬挂问题，深复制解决此问题。

（6）对象的应用：对象数组，对象指针，对象引用，对象参数。

对象作为函数参数的三种形式：传值（对象变量）、传地址（对象指针）、传名（引用）。

练习 1：以下语句调用哪种构造函数：

```
class A;    A a, b[3];          //调用不带参数的构造函数
class A;    A *p=new A;         //调用不带参数的构造函数
class A;    A *p=new A[3]; //调用不带参数的构造函数
class A;    A a, b=a;           //a 调用不带参数的构造函数
                                //b 调用复制构造函数
class A;    A a, b(a);          //a 调用不带参数的构造函数
                                //b 调用复制构造函数
class A;    A a,b; b=a;         //赋值运算
class A; void fun(A x); A a; fun(a);
//对象作为函数参数或返回值时会调用复制构造函数
//调用复制构造函数的 3 种情况！
```

（教材例 3.9，例 3.12，课后习题 3 三 1、2，实验一 1、2，MOOC 编程作业）

第 4 章：类与对象的知识进阶

需要掌握的知识要点：

（1）对象成员：（构造函数调用次序）。

（2）静态成员：静态 static 数据成员、静态成员函数（没有 this 指针）。
属于类的范畴，为同类对象所共享。

（3）常对象与常成员：（常数据成员、常成员函数+const）。

（4）友元：为了提高效率，方便编程。
友元函数（没有 this 指针），友元成员，友元类。
友元关系是单向的，不具有交换性和传递性。
前向声明的使用

练习：

1 该程序运行输出是：__3 4 3__

```
#include <iostream>
using namespace std;
class aClass {
public:
   aClass ( ) {   total++;   }
   ~aClass ( ) { total--;   }
   int gettotal( ) {   return total; }
private:
    static int total;
};
int aClass :: total = 0;
int main() {
    aClass o1,o2,o3;
    cout<<o1.gettotal()<<"    ";
    aClass *p;
    p = new aClass;
    cout<<o1.gettotal()<<"    ";
    delete p;
    cout<<o1.gettotal()<<endl;
    return 0;                          }
```

（教材例 4.2，例 4.9-4.10，课后习题 4 三-1、2，四-1，实验一 3，MOOC 编程作业）

第 5 章：继承性

需要掌握的知识要点：

（1）继承与派生的基本概念：基类（父类），派生类（子类），单一继承，多重继承。

（2）派生的定义与访问控制：不同继承方式，基类成员在派生类中的访问属性。

基类的 private 成员不能被继承（不可直接访问）。

（3）派生类的构造和析构：

构造函数调用顺序：

虚基类—>非虚基类—>派生类对象的构造函数（按定义顺序）—>派生类构造函数

构造函数调用顺序：

正好与构造函数的调用顺序相反。

（4）同名冲突及其解决方法：

单继承中的同名问题（同名覆盖）使用成员名限定法：基类名::

多重继承中的同名问题（二义性）

使用成员名限定法：类名::

使用虚基类（菱形继承）：virtual

（5）赋值兼容原则：

基类对象＝派生类对象；

基类对象指针＝派生类对象地址；

基类对象引用＝派生类对象；//初始化

基类对象指针＝派生类对象指针；

练习：

1 该程序运行输出是：__200 401 499__。

```
#include <iostream>
using namespace std;
class A
{ public:
   int x;
   A(int i) { x=i;}
   void Show() { cout<<x<<"    "; }
};
class B
{ public:
    int y;
    B(int i) { y=i; }
    void Show() { cout<<y<<"    "; }
};
class C: public A, public B
{ public:
```

```
        int y;
        C(int a, int b, int c):A(a+1), B(b-1) { y=c; }
        void Show() { cout<<y<<"    "; }
    };
    void main()
    {   C c1(400, 500, 600);
         c1.y=200;
         c1.Show();
         c1.A::Show();
         c1.B::Show();
    }
```

（教材例5.5，例5.6，例5.9，课后习题5四-1、2、3，实验二-1、2，MOOC编程作业）

第 6 章：多态性

需要掌握的知识要点：

（1）多态的两种类型：静态多态性， 动态多态性。

（2）静态多态性的实现：编译时多态性：函数（运算符）重载。

运算符重载两种方式：通过类中成员函数或类的友元函数重载。

只能重载为成员函数的运算符：“=”、 “( )”、 “[ ]”、 “->”，单目运算符及复合赋值运算符也建议重载为成员函数，第一运算对象必须本类对象

只能重载为友元函数的运算符： 提取符“>>”、插入符“<<”。

不能重载的运算符：“.” 、“.*” 、“::”、 “sizeof”、 “?:”。

（3）动态多态性的实现：运行时多态性：继承、虚函数、基类指针或引用。

（4）纯虚函数与抽象类：拥有至少一个纯虚函数的类才可以称为抽象类。抽象类不能生成对象。

重点：

（1） 运算符重载（成员函数、友元函数）

例如：class X;

```
X operator + ( X a );   //成员函数，双目运算+、-、*、/
friend X operator + ( X a, X b); //友员函数，双目运算
X operator - ( );   //成员函数，单目运算 -（取反）
friend X operator - ( X a); //友员函数，单目运算
X operator ++ ( ) //成员函数，前缀++、--
friend X operator ++ ( X& a); //友员函数，前缀++、--
X operator ++ ( int ) //成员函数，后缀++、--
friend X operator ++ ( X& a, int); //友员函数，后缀++、--
friend ostream& operator << ( ostream & out, const X &obj);
                                    //友员函数，输出运算符<<
friend istream& operator >> ( istream & in, X &obj);
                                    //友员函数，输入运算符>>
```

（教材例6.7、实验三1）

（2） 虚函数的特点、虚函数的应用、抽象类

（教材例6.12，课后习题6四-1、2，实验三2，MOOC编程作业）

第 7 章：模板

需要掌握的知识要点：

（1）模板的概念及分类：提高代码的复用率，分为函数模板，类模板。

（2）函数模板与模板函数：

（3）类模板与模板类：

（教材例 7.2、7.3、7.4、课后习题 7 三-1、2、3）

第 8 章：C++的 I/O 流类库

需要掌握的知识要点：

（1）I/O 流的概念及流类库： 输入流，输出流，输入输出流。

（2）键盘输入与屏幕输出：cin、cout。

（3）文件的输入/输出：文件为二进制文件和文本文件两种。

文件操作四个步骤：定义文件流，打开，读写，关闭文件。

3 种类型的文件流：输入文件流 ifstream，输出文件流 ofstream，输入输出文件流 fstream，都定义在名字空间 std 的 fstream 文件中。文件操作，必须在程序的开头用#include <fstream>及 using namespace std;进行文件包含。

（3）文件的读写 ：

读文件：数据从磁盘文件读取至内存

写文件：数据从内存存放至磁盘文件

文本文件读写：使用运算符“>>”、“<<”;

put( )、get( )函数。

二进制文件读写：使用 read( )、write( )函数。

（教材例 8.6，例 8.7，例 8.9，实验四）