湖南省高等教育自学考试

课 程 考 试 大 纲

面向对象程序设计

（课程代码：02328）

湖南省教育考试院组编

2021 年 12 月

高等教育自学考试课程考试大纲

课程名称：面向对象程序设计　　课程代码：02328

第一部分　课程性质与目标

一、课程性质与特点

面向对象程序设计是高等教育自学考试软件工程、自动化专业的选考课程。它是为满足从事软件工程工作人才的需要而设置的。本课程的任务是通过全面、系统地学习面向对象程序设计的基本概念、分析方法、设计方法、C++语言基本语法和编程方法；全面理解 C++语言面向对象的基本特性，包括类、对象、派生类、继承、多态性、虚函数、模板、流类库等；深刻理解和领会面向对象程序设计的特点和风格，类的封装性、继承性和多态性与程序的安全性、结构性和灵活多样性之间的关系，达到掌握其设计方法和编程基础的目的，使考生为以后学习 Java 语言程序设计、软件工程等后继课程及进行课程设计打下必备的基础，并且为以后从事应用软件开发提供合适的工具。

二、课程目标与基本要求

通过本课程的学习，要求考生：

1．了解面向对象的思想和基本概念。

2．理解 C++源程序的构成，掌握 C++程序的编辑、编译、连接和运行的方法，全面理解 C++对 C 的扩充，内容包括 C++的输入输出， C++灵活的局部变量说明，const 修饰符的使用，函数原型的作用及使用方法，内联函数的作用及定义方法，带有默认参数的函数及默认参数定义方法，函数重载作用及重载方法，作用域运算符“::”， 强制类型转换，运算符 new 和 delete 及引用数据类型作用及定义方法。

3．理解和掌握面向对象的核心概念——类和对象。能定义类，并利用类创建对象，通过访问对象的成员实现对象的联系。能熟练利用类和对象进行编程。

4．设计合理的类层次，并使用继承和派生构造应用程序。

5．理解多态性，掌握实现多态性的方法及其在面向对象中的重要意义。

6. 理解运算符重载的方法及使用，掌握不同类型数据间实现转换方法。

7. 理解模板的概念，掌握函数模板和类模板的定义、模板函数和模板类使用方法。

8.了解 C++流类库的基本结构，掌握预定义类型输入输出的格式控制方法，掌握通过文件流实现文件的操作的方法。

9. 了解 C++异常处理机制，掌握利用异常处理机制实现在环境出现意外的情况下，作出正确合适的处理和防范，了解命名空间的作用和使用方法，并掌握处理程序中同名冲突问题的方法

10.熟练使用 C++语言采用面向对象方法进行应用程序的开发。

要求考生能切实掌握 C++程序设计语言作为实际工作中的工具，并为以后学习其它课程打下基础，本课程是一门实践性很强的课程，要求考生不仅要掌握编程方法，而且能在计算机上调试和测试一般的程序。

三、与本专业其他课程的关系

本课程的先行课程为计算机导论，以便对计算机软硬件相关概念有一个初步的了解。本课程的后继课程是 Java 语言程序设计等，本课程学习的面向对象程序设计的思想和基础概念帮助考生学习 Java 语言程序设计。学好本门课程同样有助于课程设计。

第二部分　考核内容与考核目标

第一章　面向对象方法概述

一、学习目的与要求

通过本章学习，了解面向过程程序设计方法及其局限性，理解面向对象程序设计方法的基本概念、基本特征、主要优点。

二、考核知识点与考核目标

（一）面向对象程序设计方法的基本概念、基本特征、主要优点（重点）

识记：1. 对象、类、消息、方法的概念

2. 面向对象程序设计的基本特征

理解：1. 类与对象的关系

2. 数据封装、继承、多态性

第二章　C++的初步知识

一、学习目的与要求

本章介绍 C++的发展和特点、C++源程序的构成、 C++程序的编辑、编译、连接和运行及 C++对 C 的扩充等。通过本章学习，理解 C++对 C 的面向过程程序设计扩充的内容，能编写完成简单的面向过程的 C++应用程序。

二、考核知识点与考核目标

（一）C++的发展和特点，C++源程序的构成， C++程序的编辑、编译、连接和运行（次重点）

识记：1. C++与 C 语言的关系

2.源程序、程序的编辑、编译、连接和运行等概念

理解：1. C++源程序的结构

2.main( )函数的作用

（二）C++对 C 的扩充（重点）

识记：1. C++注释、const 函数原型、内联函数、函数的重载

2.数据类型转换的两种方式

3.标识符作用域、内存动态分配与回收、引用等概念

理解：1.C++输入输出对象及插入、提取运算符、数据输入方式

2.const 的作用及使用方法

3. 函数原型的作用、内联函数的作用，作用域运算符“::”的作用

4.动态数组分配与回收方法、动态变量初始化方法、引用的定义方法

5.区分变量名、指针变量及引用作为函数参数的不同的作用

应用：1.函数原型的使用方法、内联函数的使用方法

2. 利用 new 和 delete 实现内存动态管理

3. 函数带默认参数的定义方法、函数重载的实现方法

第三章　类与对象

一、学习目的与要求

通过本章学习，理解类的概念及其作用、类的定义，能用类声明与创建对象，掌握访问对象成员的方法、对象初始化及其析构，掌握使用类和对象进行编程。

二、考核知识点与考核目标

（一）定义类、对象的创建和对象的初始化、使用对象成员（重点）

识记：1.标识符的类作用域

2.类成员的访问属性、信息隐蔽、数据保护

3.构造函数、析构函数、默认的构造函数、默认的析构函数

理解：1.类成员函数的定义

2.构造函数及构造函数的两种调用方式

3.用成员初始化表对数据成员进行初始化的方法

4.带默认参数的构造函数、构造函数重载的方法

5.对象的生命周期及析构函数的调用时机

应用：1.声明类的方法、类成员的访问控制

2.对象声明，对象成员初始化的方法

3.使用对象成员

（二）对象的赋值与复制、自引用指针 this, C++的 string 类（一般）

识记：1.拷贝构造函数

2.自引用指针 this, C++的 string 类

理解：1.默认的拷贝构造函数实现的复制方式

2. 调用拷贝构造函数的 3 种情况

3. 利用 string 类创建对象、string 类常用运算符

应用：自定义拷贝构造函数的方法

第四章　类和对象的进一步讨论

一、学习目的与要求

通过本章学习，掌握对象数组的声明、对象数组元素的初始化方法与使用方法、对象指针的声明、用对象指针访问单个对象和对象数组的方法，学会利用对象数组组织相同类型的对象，理解向函数传递对象、对象指针、对象引用的区别，了解静态成员作用、静态成员的特性、静态数据成员的定义和初始化方法、静态成员函数的定义和在静态成员函数中访问其它成员的特性。了解友元的作用，掌握友元函数的声明和友元类的声明方法。掌握用已有类组合出新类的方法。掌握用 const 修饰符来保证共享数据不能被改动，理解类声明，类实现各自的作用。

二、考核知识点与考核目标

（一）对象数组、对象指针、向函数传递对象、 静态成员（重点）

识记：1. 对象数组、对象指针

2. 静态数据成员、静态成员函数

理解：1 对象数组的定义与调用无参数、一个参数及多个参数构造函数实现数组初始化的方法

2.对象指针的定义，用对象指针访问单个对象与对象数组的方法

3.能区分对象做函数参数，对象指针做函数参数，对象引用做函数参数的区别

4.静态数据成员的定义、初始化、特性及访问方法

5.静态成员函数的定义与作用、对数据成员的访问特性、访问静态成员函数的方法

应用：1.利用对象数组、对象指针对多个同类型对象进行组织管理

2.对象共享的成员定义静态成员，并用静态成员函数来访问

（二）友元、类的组合、共享数据的保护（次重点）

识记：1.友元的作用、friend 的含义、友元函数和友元类

2.类的组合关系

3.常对象、常数据成员、常成员函数

理解：1.友元函数的声明和友元类的声明

2.对象成员的定义、使用初始化列表实现对象成员初始化

3.带对象成员的类的构造函数的调用顺序

4.常对象、常数据成员、常成员函数的声明，使用初始化列表实现常数据成员初始化

应用：编程实现利用友元函数访问对象的私有成员

（三）C++的多文件程序（一般）

识记：类声明、类实现 *.h,*.cpp 文件

第五章　继承与派生

一、学习目的与要求

通过本章学习，掌握单一继承、多重继承、两义性、支配规则和虚基类的概念，掌握派生类的访问权限，构造函数与析构函数的调用顺序，理解多重继承两义性产生的原因及处理方法，能熟练运用作用域运算符，掌握虚基类的定义及其作用。

二、考核知识点与考核目标

（一）继承的概念、继承的实现方法、继承成员的访问控制规则、类型兼容性规则、派生类对象继承自基类的成员初始化的方法和析构方法等（重点）

识记：1.继承的概念及实现方式

2.基类和派生类的概念

3.继承成员的访问控制规则，访问控制声明

理解：1.派生类对象继承自基类的成员初始化的方法和析构方法

2.父子类构造函数与析构函数的调用顺序

3.子类成员函数重定义父类的成员，子类对象访问的是重定义后的成员函数，若要访问父类成员应用基类名::成员名

4.含有对象成员的派生类的构造函数的表达方式及执行顺序

应用：利用继承机制实现类层次编程

（二）多重继承、多重继承的名字冲突问题及解决方法、虚基类及其作用（次重点）

识记：1.多重继承的实现方法

2.虚基类的定义方法

理解：1.多重继承的名字冲突问题及解决方法

2.虚基类的作用及虚基类成员初始化方法

第六章　多态性与虚函数

一、学习目的与要求

通过本章学习，深刻理解多态性、编译时多态性、运行时多态性，纯虚函数、抽象类等概念，深刻理解多态性在面向对象程序设计中的重要意义，掌握基类与派生类对象之间的赋值兼容关系，熟练掌握实现静态多态性和动态多态性的方法。理解纯虚函数、抽象类的含义及特性。

二、考核知识点与考核目标

（一）多态性多态性与虚函数（重点）

识记：多态性、编译时多态性、运行时多态性、纯虚函数、抽象类等概念

理解：1.父子类间类型兼容性规则

2.函数调用的两种绑定方式

3.虚函数和纯虚函数声明方式，运行时多态性出现的条件，虚函数的执行与虚析构函数的执行

4.纯虚函数的定义、抽象类的特性

应用：如何用虚函数实现运行时多态性

第七章　运算符重载

一、学习目的与要求

运算符重载是面向对象程序设计的重要特性，通过本章学习，理解运算符重载的方法及使用，掌握不同类型数据间实现转换方法。

二、考核知识点与考核目标

（一）运算符重载、不同类型数据间实现转换（一般）

识记：1.运算符重载的两种方式

2.关键字 operator 的含义

3.重载运算符时应该遵守的规则

理解:　1.运算符友元重载函数和成员重载函数在参数上的区别

2.重载“<<”和“>>”来实现自已类型的输入 / 输出操作

3.通过转换构造函数和类型转换函数进行类型转换

第八章 模板

一、学习目的与要求

通过本章学习，深刻理解模板的概念，掌握函数模板和类模板的定义和使用。

二、考核知识点与考核目标

（一）C++语言使用模板来实现类型参数化、函数的模板和类模板使用（次重点）

识记：模板、函数模板、类模板、template 关键字、模板函数、模板类的概念

理解：1.定义函数模板和定义类模板的方法

2.模板函数的实例化和模板类的实例化及使用

3. C++编译器在匹配函数时的约定

应用：能设计函数模板，并能实例化函数模板为模板函数，并能调用它。

第九章　C++的输入和输出

一、学习目的与要求

通过本章学习，了解 C++流类库的基本结构，掌握预定义类型输入输出的格式控制方法，掌握通过文件流实现文件的操作的方法。

二、考核知识点与考核目标

（一）C++标准输入输出流的使用、数据格式化输入 / 输出、文件流的用法（次重点）

识记：1.C++语言的输入 / 输出，C++流类库

2.格式化输入 / 输出，输入输出操作符

3.流类库提供的常用成员输入/输出函数，文件流及各种文件操作

理解：文本文件的读写过程及方法

第十章　异常处理和命名空间

一、学习目的与要求

通过本章学习，了解C++异常处理机制，掌握利用异常处理机制实现在环境出现意外的情况下，作出正确合适的处理和防范，了解命名空间的作用和使用方法，并掌握处理程序中同名冲突问题的方法。

二、考核知识点与考核目标

（一）异常处理（次重点）

识记：异常、异常处理、异常检查、异常抛出和异常捕获

理解：1.C++处理异常的办法，异常的检查和捕获方法

2.发生异常时，程序的执行流程

（二）命名空间和头文件命名规则（次重点）

识记：命名空间、C++头文件命名规则

理解：1.命名空间声明方法

2.使用命名空间中的标志符的两种方式

第十一章　综合设计与实现

（不作考核要求）

第三部分　有关说明与实施要求

一、考核的能力层次表述

本大纲在考核目标中，按照“识记”、“理解”、“应用”三个能力层次规定其应达到的能力层次要求。各能力层次为递进等级关系，后者必须建立在前者的基础上，其含义是：

识记：能知道有关的名词、概念、知识的含义，并能正确认识和表述，是低层次的要求。

理解：在识记的基础上，能全面把握基本概念、基本原理、基本方法，能掌握有关概念、原理、方法的区别与联系，是较高层次的要求。

应用：在理解的基础上，能运用基本概念、基本原理、基本方法联系学过的多个知识点分析和解决有关的理论问题和实际问题，是最高层次的要求。

二、教材

指定教材：c++面向对象程序设计，陈维兴、陈昕编著，人民邮电出版社，2010年10月第1版；

三、自学方法指导

1．在开始阅读指定教材某一章之前，先翻阅大纲中有关这一章的考核知识点及对知识点的能力层次要求和考核目标，以便在阅读教材时做到心中有

数，有的放矢。

2．阅读教材时，要逐段细读，逐句推敲，集中精力，吃透每一个知识点，对基本概念必须深刻理解，对基本理论必须彻底弄清，对基本方法必须牢固掌握。

3．在自学过程中，既要思考问题，也要做好阅读笔记，把教材中的基本概念、原理、方法等加以整理，这可从中加深对问题的认知、理解和记忆，以利于突出重点，并涵盖整个内容，可以不断提高自学能力。

4．完成书后作业和适当的辅导练习是理解、消化和巩固所学知识，培养分析问题、解决问题及提高能力的重要环节，在做练习之前，应认真阅读教材，按考核目标所要求的不同层次，掌握教材内容，在练习过程中对所学知识进行合理的回顾与发挥，注重理论联系实际和具体问题具体分析，解题时应注意培养逻辑性，针对问题围绕相关知识点进行层次（步骤）分明的论述或推导，明确各层次（步骤）间的逻辑关系。

四、对社会助学的要求

1．应熟知考试大纲对课程提出的总要求和各章的知识点。

2．应掌握各知识点要求达到的能力层次，并深刻理解对各知识点的考核目标。

3．辅导时，应以考试大纲为依据，指定的教材为基础，不要随意增删内容，以免与大纲脱节。

4．辅导时，应对学习方法进行指导，宜提倡“认真阅读教材，刻苦钻研教材，主动争取帮助，依靠自己学通”的方法。

5．辅导时，要注意突出重点，对考生提出的问题，不要有问即答，要积极启发引导。

6．注意对考生能力的培养，特别是自学能力的培养，要引导考生逐步学会独立学习，在自学过程中善于提出问题，分析问题，做出判断，解决问题。

7．要使考生了解试题的难易与能力层次高低两者不完全是一回事，在各个能力层次中会存在着不同难度的试题。

8．助学学时：本课程共 3 学分，建议总课时 48 学时，其中助学课时分配如下：

章 次	内 容	学 时
第一章	面向对象方法概述	2
第二章	C++的初步知识	4
第三章	类和对象	6
第四章	类和对象的进一步讨论	6
第五章	继承与派生	6
第六章	多态性与虚函数	6
第七章	运算符重载	4
第八章	模板	6
第九章	C++的输入和输出	4

第十章	异常处理和命名空间	4
合　　计		48

五、关于命题考试的若干规定

1．本大纲各章所提到的内容和考核目标都是考试内容。试题覆盖到章，适当突出重点。

2．试卷中对不同能力层次的试题比例大致是：“识记”为 20%、“理解”为 40%、“应用”为 40%。

3．试题难易程度应合理：易、中等、难比例为 3：4：3。

4．每份试卷中，各类考核点所占比例约为：重点占 60%，次重点占 30%，一般占 10%。

5．试题类型一般分为：单项选择题、填空题、程序改错题、程序填空题、程序分析题、程序设计题。

6．考试采用闭卷笔试，考试时间 150 分钟，采用百分制评分，60 分合格。

六、题型示例（样题）

一、单项选择题（本大题共■小题，每小题■分，共■分）

在每小题列出的四个备选项中只有一个是符合题目要求的，请将其选出并将“答题卡”上的相应字母涂黑。错涂、多涂或未涂均无分。

1．所谓多态性是指

A．不同的对象调用不同名称的函数　　B．不同的对象调用相同名称的函数

C．一个对象调用不同名称的函数　　D．一个对象调用不同名称的对象

2．以下有关继承的叙述正确的是

A．构造函数和析构函数都能被继承

B．派生类是基类的组合

C．派生类对象除了能访问自己的成员以外，不能访问基类中的所有成员

D．基类的公有成员一定能被派生类的对象访问

二、填空题（本大题共■小题，每小题■分，共■分）

1．对虚函数使用对象指针或引用，系统使用_____联编，对虚函数使用对象调用时，系统使用_____联编。

2．静态成员定义的关键字为____，一般通过_____来访问静态成员。

三、程序改错题（本大题共■小题，每小题■分，共■分）

1．下面程序中有一处错误，请用下横线标出错误所在行并说明出错原因。

```
class base{
protected:
int p;
public:
Base(int m){p=m;}
};
void f()
{   Base a(10);
```

```
    cout<<a.p<<endl;
}
```

2．下面程序中有一处错误，请用下横线标出错误所在行并说明出错原因。

```
class Base{
public:    virtual void fun()=0;
};
class Test: public Base{
public: virtual void fun(){cout<<"Test.fun="<<endl;}
};
void main() {
Base a;
Test *p;
p=&a;
}
```

四、程序填空题（本大题共■小题，每小题■分，共■分）

1．完成下面类中的成员函数的定义。

```
class   test   {
    private:
    int num;
    float f1;
public:
    test(int , float f);
    test(test&);
    }
test::test(              )
    {
    num=n;
    =f;
    }
test::test(            )
    {
num=
    f1=t.f1
    }
```

2．在下面一段类定义中，Derived 类公有继承了基类 Base。需要填充的函数由注释内容给出了功能。

```
class Base
{
     private:
int mem1,mem2;          //基类的数据成员
     public:
Base(int m1,int m2) {
     mem1=m1; mem2=m2;
```

```
}
    void output(){cout<<mem1<<’ ’<<mem2<<’ ’;}
    //...
};

class Derived: public Base
{
    private:
        int mem3;      //派生类本身的数据成员
public:
    //构造函数,由 m1 和 m2 分别初始化 mem1 和 mem2，由 m3 初始化 mem3
    Derived(int m1,int m2, int m3);
    //输出 mem1,mem2 和 mem3 数据成员的值
    void output(){
;
cout<<mem3<<endl;
    }
  //...
};
Derived::Derived(int m1,int m2, int m3):   (2)
{            (3)  ;       }

(1)                          (2)                          (3)
```

五、程序分析题（本大题共■小题，每小题■分，共■分）

分析程序，写出运行结果。

1．
```
#include<iostream.h>
void swap(int &,int &);
void main( )
{      int a=5,b=8;
         cout<<"a="<<a<<","<<"b="<<b<<endl;
swap(a,b);
         cout<<"a="<<a<<","<<"b="<<b<<endl;
}
void swap(int &x,int &y)
{int temp=x;      x=y;      y=temp;}
```

2．
```
#include<iostream.h>
class a
{
public:
virtual void print(){cout<< " this is class a printing. " << endl; };
};
class b: public a
```

```
{
public:
void print(){ } ;
};
class c: public b
{
public:
void print(){cout<< " this is class c printing. " <<endl;}
};
void show(a &aa)
{
aa.print();
}
void main()
{
a a;
b b;
c c;
show(a);
show(b);
show(c);
    }
```

六、程序设计题（本大题共■小题，每小题■分，共■分）

根据要求，编写程序。

1．声明一个 Circle 类，有数据成员 radius(半径)、成员函数 area(),计算园的面积，构造一个 Circle 类的对象进行测试。

2．设计一个模板函数 max_of_array(),该函数从一个数组中找出其中的最大元素，数组中存放元素的类型可能是 int、float、double 之一，并编写 main()测试该模板函数。