

# 《现代 C++ 程序设计》课程大纲

## 一、课程简介

本课程面向软件工程专业开设，为专业选修课，是掌握高级程序设计语言比较深入的课程。课程引入 C++ 23 标准，详细讲解面向对象程序设计的思想，包括面向对象的核心概念：封装性、继承性和多态性，以及泛型编程、多线程等，也包含针对具体问题的编程实例。

## 二、课程模块

### 1. CM1: C++新特性和面向对象基础

C++ 23 语言的新概念及一些新特点：左值引用和右值引用、基于区间的 for 语句、constexpr 和 consteval、if consteval 语句、lambda 表达式、异常处理语句、函数重载等。类的概念、语法及其高级特性：构造函数、析构函数、this 指针、友元等。

### 2. CM2: C++高级编程

运算符重载的原理、原则和方法；类的继承的原理、原则和方法；赋值兼容原则；多继承；运行时的多态，包括：虚函数的概念、原理和实现方法、纯虚函等。

### 3. CM3: C++复杂应用编程

C++ 模板：变量模板、函数模板和类模板的概念和设计方法，以及它们的高级特性；C++ 的容器的概念及特点；迭代器的概念和原理；泛型算法（库）的原理和设计方法；concept 的概念和设计方法；多线程和协程的基础和设计方法。

## 三、课程内容和要求

### （一）教学内容和要求

本课程共 32 学时，课堂讲授 32 学时，内容包括（其中学时为建议学时，可微调）：

#### 第 1 章 引论（CM1），0.5 学时

**教学内容：**对象的概念、面向对象技术的核心概念，C++ 程序的编辑、编译和链接方法。

- (1) 对象的概念、面向对象程序设计的基本思想。
- (2) 对象的概念
- (3) 面向对象技术的核心概念、C++ 程序开发的步骤。

**重点：**

- (1) 对象的基本概念

(2) 面向对象技术的核心概念

(3) 指针

**难点：**

(1) 用对象的思想进行设计

(2) 面向对象与面向过程两种方法的不同

**教学要求：**

掌握：C++程序的编译、链接和运行过程；对象的概念。

理解：面向对象的核心概念；面向对象与面向过程的区别。

了解：无。

## **第2章 C++：一个更好的C（CM1），0.5学时**

**教学内容：**C++新的基础类型；C++引用类型；类型自动推导；lambda表达式；基于范围的for语句；异常处理及语句；C++函数，包括函数重载的概念和实现机制；复杂类型声明的简化；名字空间；C++模块。

**重点：**

(1) C++新的语法成分

(2) 引用类型，包括左值引用和右值引用

(3) Lambda表达式

(4) 异常处理及语句

(5) 函数重载

**难点：**

(1) 引用类型，尤其是右值引用

(2) Lambda表达式

**教学要求：**

掌握：C++新的语法成分，包括引用类型、lambda表达式、异常处理；函数重载方法。

理解：类型自动推导；名字空间。

## **第3章 类：面向对象的基石（CM1），4学时**

**教学内容：**面向对象的程序设计方法是运用面向对象的观点对现实世界的各种问题进行抽象，并用计算机程序来描述并解决问题。这种描述和处理通过类与对象来实现。本讲介绍类与对象的基本概念。

- (1) 数据抽象与封装的概念；类的概念和定义。
- (2) 对象的概念；类与对象之间的关系；对象成员的使用方法。
- (3) 类的访问控制
- (4) 静态数据成员和静态成员函数
- (5) C++的类
- (6) 类对象的构造、初始化和析构：构造函数及重载、统一初始化、析构函数

**重点：**

- (1) 类与对象的概念；类的定义；对象的定义。
- (2) 静态成员。
- (3) 构造函数及重载、析构函数

**难点：**

- (1) 类的静态成员
- (2) 类的构造函数重载

**教学要求：**

掌握：类的概念、定义方法；类的访问控制、类成员的访问属性；对象的定义；对象成员的访问；静态成员的特点与作用；类的构造函数、析构函数的概念、语法、用途；掌握数据封装的概念

理解：封装机制；通过类对抽象结果进行描述；类与对象的关系。

了解：如何使用面向对象方法编写应用程序。

#### **第4章 类的高级特性（CM1），4学时**

**教学内容：**

- (1) 复制控制：复制、赋值；深复制和浅复制；转移对象和转移语义；禁止复制
- (2) 指向类成员的指针
- (3) 友元：友元函数和友元类
- (4) 类的 const 成员和 mutable 成员
- (5) 类中的类型名

**重点：**

- (1) 复制控制
- (2) 友元
- (3) 类中的类型名

**难点:**

- (1) 复制控制
- (2) 转移语义

**教学要求:**

**掌握:** 复制控制的概念、复制构造函数的设计; 转移语义及其实现方法; 友元的概念、使用方法; 类中类型及其别名的定义方法。

**理解:**

**了解:** 赋值运算符函数的重载

## **第 5 章 运算符重载 (CM1), 3 学时**

**教学内容:** 本讲主要讲述 C++ 运算符重载的思想和方法, 内容包括运算符的重载形式、常用运算符的重载, 并给出实际案例。

- (1) 运算符重载形式和规则
- (2) 常用运算符的重载
- (3) 类型转换
- (4) 重载特殊运算符

**重点:** 常用运算符的重载。

**难点:** []和()运算符的重载。

**教学要求:**

**掌握:** 运算符重载的概念和意义; 运算符重载的原理、语法; 运算符重载的方法; 常用运算符的重载

**理解:** 运算符函数参数和返回的选择

**了解:** 特殊运算符重载

## **第 6 章 继承 (CM2), 4 学时**

**教学内容:** 本讲介绍继承与派生的相关概念及其在面向对象程序设计中的应用。

- (1) 软件重用思想; 继承与派生的概念

- (2) 继承和派生：派生类的定义；派生类的构造函数与析构函数
- (3) 赋值兼容原则
- (4) 多继承
- (5) 继承的前提：正确的分类

**重点：**

- (1) 派生类的定义
- (2) 派生类的构造与析构
- (3) 继承的实现机制
- (4) 赋值兼容原则
- (5) 正确的分类

**难点：**

- (1) 赋值兼容原则
- (2) 正确的分类

**教学要求：**

掌握：继承与派生的概念；赋值兼容原则；正确的分类

理解：面向对象程序设计如何实现软件重用

了解：多继承

## **第7章 多态（CM2），1学时**

**教学内容：**本讲介绍面向对象程序设计多态性的概念及相关知识。

- (1) 多态的概念
- (2) 虚函数：实现多态的关键
- (3) 纯虚函数和抽象类

**重点：**

- (1) 多态的思想与实现形式
- (2) 虚函数实现原理
- (3) 纯虚函数与抽象类的概念及应用。

**难点：**

- (1) 虚函数的实现机制；
- (2) 抽象类的继承

**教学要求：**

掌握：多态的概念；虚函数的概念和语法。纯虚函数与抽象类的概念及应用。

理解：多态的实现机制

**第 8 章 模板（CM3），6 学时**

**教学内容：**本讲介绍模板与泛型编程的相关概念。

- (1) 变量模板
- (2) 函数模板
- (3) 类模板
- (4) 模板的别名
- (5) traits 技术
- (6) 模板元编程初探

**重点：**

- (1) 函数模板与模板函数
- (2) 类模板与模板类
- (3) traits 技术

**难点：**

- (1) 模板的设计
- (2) traits 技术

**教学要求：**

掌握：函数模板的概念及其语法；类模板的概念及语法。

理解：泛型程序设计的思想

了解：traits 的设计思路。

**第 9 章 容器、迭代器和泛型算法（CM3），6 学时****教学内容：**

- (1) 容器和迭代器的概念、结构和设计。
- (2) 泛型算法
- (3) 概念(Concept)、范围(range)和视图(view)
- (4) STL 库

**重点：**

- (1) 迭代器的概念和设计
- (2) 泛型算法的设计

**难点：** 迭代器的概念和设计

**教学要求：**

掌握：容器和迭代器的概念；迭代器的结构和设计；泛型算法的设计。

理解：STL 库

## **第 10 章 多线程（CM3），3 学时**

**教学内容：**

- (1) 相关关键概念。
- (2) C++的多线程库：thread、jthread、mutex、conditional\_variable、semaphore、latch、barrier、future、promise、协程等
- (3) 多线程编写方法

**重点：**

- (1) 多线程的相关关键概念
- (2) thread
- (3) mutex

**难点：** 多线程编程

**教学要求：**

掌握：无

理解：C++的多线程库

了解：多线程相关的关键概念

## **（二）自学内容和要求**

### **第 11 章 面向对象设计的原则**

**教学内容：**

面向对象设计的 7 大原则：单一职责原则、开闭原则、聚集与组合复用原则、里氏替换原则、依赖倒置原则、接口隔离原则、最少知识原则

**重点：** 单一职责原则、开闭原则、聚集与组合复用原则、里氏替换原则

**难点：** 依赖倒置原则、接口隔离原则、最少知识原则

**教学要求：**

**掌握：** 单一职责原则、开闭原则、聚集与组合复用原则、里氏替换原则

**了解：** 依赖倒置原则、接口隔离原则、最少知识原则

## **第 12 章 C++程序设计案例**

**教学内容：**

一个复杂的 C++程序设计案例：MVC 设计模式、复杂应用编码

**重点：** MVC 设计模式

**难点：** 复杂应用编码

**教学要求：**

**掌握：** MVC 设计模式的概念

**了解：** 复杂应用的设计方法

## **四、 建议教材及参考资料**

### **（一） 教材**

《C++程序设计——现代方法》，白忠建编著，人民邮电出版社，2019 年 12 月。ISBN：978-7-115-51373-1。

### **（二） 参考资料**

1. C++程序设计与实践（第二版），白忠建编著，机械工业出版社，2016 年 10 月。
2. C++程序设计语言（特别版）十周年中文纪念版.[美]Bjarne Stroustrup 著，机械工业出版社, 2010.3
3. C++编程思想. [美] Bruce Eckel 著，机械工业出版社，2000.1
4. C++ Primer（中文版，第五版）.[美]Stanly B. Lippman等著，王刚，杨巨峰译，电子工业出版社，2013.9