



《面向对象程序设计 C++》教学大纲

2017 年 5 月 15 日

目 录

第一部分 大纲说明	4
第二部分 教学设计	5
2.1 教学手段	5
2.2 授课思路	5
2.3 学时分配	6
2.3 课程考核	6
第三部分 目标细化	7
3.1 C++语言概述	7
3.1.1 主要内容	7
3.1.2 教学目标	7
3.1.3 重点难点	7
3.2 从 C 到 C++	8
3.2.1 主要内容	8
3.2.2 教学目标	8
3.2.3 重点难点	9
3.3 类与对象（上）	10
3.3.1 主要内容	10
3.3.2 教学目标	10
3.3.3 重点难点	11
3.4 类与对象（下）	12
3.4.1 主要内容	12
3.4.2 教学目标	12
3.4.3 重点难点	12
3.5 运算符重载	14
3.5.1 主要内容	14
3.5.2 教学目标	14
3.5.3 重点难点	14
3.6 继承	13
3.6.1 主要内容	13
3.6.2 教学目标	13
3.6.3 重点难点	13
3.7 多态	15
3.7.1 主要内容	15
3.7.2 教学目标	15
3.7.3 重点难点	15
3.8 动态数组（线性表的顺序存储）	16

3.8.1 主要内容.....	16
3.8.2 教学目标.....	16
3.8.3 重点难点.....	16
3.9 链表（线性表的链式存储）	17
3.9.1 主要内容.....	17
3.9.2 教学目标.....	17
3.9.3 重点难点.....	17
3.10 二叉树（二叉树的二叉链表存储）	18
3.10.1 主要内容.....	18
3.10.2 教学目标.....	18
3.10.3 重点难点.....	18
第四部分 教学相关资料.....	19
教材.....	19
教学参考书及网站.....	19

第一部分 大纲说明

软件学院以专业建设为龙头，紧密跟踪软件技术的发展方向，以优质课程建设为抓手，努力培养理论基础扎实、编程技术过硬、职业素养优秀的高级软件人才。本教学大纲是依据河北师范大学软件学院软件工程专业的近三年的教学实践及课程负责人员进行的企业调研而制定。

本教学大纲适用于河北师范大学软件学院软件工程专业的本科生教学。

《面向对象程序设计 C++》是计算机科学与技术、软件工程等专业的一门重要的专业基础课。对高级语言和程序设计的学习和训练看成是计算科学专业的基本功之一，C++是一种高效而又实用的程序设计语言，它既可以进行过程化程序设计，也可以进行面向对象程序设计，因此成为了编程人员最广泛使用的工具。通过该课程的学习，学生应该掌握结构化和面向对象程序设计中基本的概念、方法，深入理解面向对象程序设计语言的三大特征（封装、多态、继承）、在理论、实践上为后续软件课程（如：Java、Python 等）打下坚实基础。

通过本课程的学习，要求学生达到下列基本目标：

知识要求	(1) 了解C++语言的发展历史，懂得面向对象语言的概念内涵。
	(2) 知道C++语言的基础知识，包括数据类型、变量等。
	(3) 学会C++语言程序的流程方面的知识、C++函数等。
	(4) 掌握C++语言的对象的概念、类的概念、及如何抽象、封装对象。
	(5) 掌握C++语言的继承、多态及应用。
	(6) 掌握使用C++描述“数据结构及算法”的方法。
	(7) 能够使用C++封装常用的“数据结构”。
	(8) 提升数据结构设计及实现相关算法的能力。
能力要求	(1) 熟练运用开发工具进行编辑、编译、调试程序。
	(2) 能够对英语的调试信息进行识别并进行有效调试。
	(3) 熟练掌握C++语言的语句。
	(4) 熟练运用C++语言的解释一些常见的算法。
	(6) 熟练运用C++语言的相关知识处理简单的实际问题。
	(7) 能够运用C++语言完成一个中等程度的实际项目。

第二部分 教学设计

2.1 教学手段

教学手段分两种：理论教学和实践教学。理论教学在课堂上完成，使用多媒体的教学方式展示借助短小精悍的示例代码，突出介绍重要的概念、重要的思想和重要的方法。实践教学在课上、课下都会进行，实践教学形式分为课程实验和课程设计两种。

课程实验要求学生根据实验手册中的实验要求，完成相应的编码、调试和测试工作，对理论教学中的方法和思想进行模仿和复现，达到巩固理论概念、强化程序设计思想和方法以及加强编程技能的目的。

课程设计则通过一个相对完整的开发需求作为典型项目开发案例，给学生提供一个能将所学到的理论知识转化为实际编码能力的途径，把课程中涉及的知识和技巧能够进行综合的运用。

2.2 授课思路

就知识点而言，比较难以掌握的知识有：指针、面向对象的三大特征（封装、继承、多态）。指针是一个与内存相关的概念、三大特征的讲解涉及到对象在内存中的存储。因此，本课程的授课思想是：主要知识点围绕着内存展开，具体而言：

- 数据类型：数据在内存中的存储方式、数据类型转换的实质及转换带来的隐患。
- 指针：指针、所指向的变量在内存中的关系示意图；
- 对象：对象在内存中是如何存储的？
- 封装：普通成员变量、静态成员变量、const 成员变量、成员函数如何在内存中存储？
- 继承：从内存的角度而言，公有继承、私有继承、保护继承的派生类对象与基类对象的关系。
- 多态：从内存的角度而言，多态是如何实现的？为什么需要多态？

在上述知识的基础上，再展开其他知识的讲授。

具体授课的方式：提出问题→解决问题→知识归纳→问题深化（拓展思考）。

2.3 学时分配

本课程总学时为 72 学时，共 3 学分。其中理论教学 36 学时，实践教学 36 学时。这两部分教学课时分配情况分别如下。

序号	主要内容	学时分配	理论课时	实验课时	作业量(次)	备注
1	C++语言概述	2	1	1	0	
2	从C到C++	8	4	4	2	
3	类与对象（上）	10	5	5	2	
4	类与对象（下）	6	3	3	2	
5	继承	10	5	5	2	
6	运算符重载	4	2	2	2	
7	多态	8	4	4	2	
8	动态数组	8	4	4	2	
9	链表	8	4	4	2	
10	二叉树	8	4	4	2	
合计		72	36	36	18	

2.3 课程考核

期末考核采用期末笔试加上机考核方式。笔试为闭卷。

本课程的成绩由三部分组成：

- 1、期中考试成绩：20 分（上机比率 70%，基础知识 30%）
- 2、平时成绩：30 分（作业、出勤、纪律、综合表现等）
- 2、期末考试成绩：50 分（上机比率 70%、基础知识 30%）

第三部分 目标细化

3.1 C++语言概述

3.1.1 主要内容

- 课程简介
- C++概述
- C 与 C++的关系
- 面向对象程序设计
- 第一个 C++程序

3.1.2 教学目标

- 了解 C++的起源
- 了解 C++的应用领域
- 理解对象和类的基本概念
- 理解面向对象的三大特征

3.1.3 重点难点

- 对象和类的基本概念
- 面向对象的三大特征

3.2 从 C 到 C++

3.2.1 主要内容

- 新的初始化方法
- C++中的输入输出
- 名字空间
- C++与 C 有关类型的区别
- 函数原型声明的区别
- 函数局部变量定义的位置
- 域解析::扩大全局变量的见范围
- 带默认参数的函数
- 内联函数
- 函数重载
- new 和 delete
- 普通变量与引用
- 指针变量的引用
- const 与引用
- 引用作为函数参数
- C++中类型转换

3.2.2 教学目标

- 理解赋值和初始化的区别
- 了解 C++中的注释的写法
- 了解 C++中的输入输出的写法
- 理解命名空间的概念
- 了解 C++中的数据类型
- 掌握 C++中 const 的使用
- 理解 C++中函数与 C 语言中函数的不同

- 掌握函数重载的用法
- 掌握引用的使用
- 掌握引用作为函数参数的使用方法
- 了解 **C++**中的强制类型转换

3.2.3 重点难点

- **C++**中的输入输出
- **string** 类型的使用
- 带默认参数的函数
- 函数重载
- **new** 和 **delete**
- 引用作为函数参数

3.3 类与对象（上）

3.3.1 主要内容

- 数据抽象与类
- 类的声明与定义
- 对象的声明与成员访问
- 成员变量与成员函数
- 访问控制与数据隐藏
- 成员初始化列表
- 对象的实例化
- 构造函数和析构函数
- 拷贝构造函数
- 静态成员
- 对象成员
- `const` 成员
- 类与结构（`struct`）的关系
- 类的作用域与可见性

3.3.2 教学目标

- 理解传统的结构化程序设计和面向对象程序设计的基本概念
- 理解类的定义,属性与行为,访问权限控制
- 掌握构造函数与析构函数的定义
- 掌握拷贝构造函数的定义，深拷贝和浅拷贝
- 掌握用初始化列表完成类成员的初始化赋值
- 掌握对象的定义,使用及内存的分配
- 掌握类如何支持封装和信息隐藏
- 使用默认实参定义构造函数
- 理解由构造函数建立对象，析构函数销毁对象的机制

- 掌握静态成员和对象成员的使用
- 掌握 `const` 与类的结合使用
- 掌握类与结构的异同

3.3.3 重点难点

- 面向对象程序设计的基本概念封装的概念
- 类的定义,属性与行为,访问权限控制
- 对象的定义,使用及内存的分配
- 拷贝构造函数的使用, 以及深拷贝与浅拷贝

3.4 类与对象（下）

3.4.1 主要内容

- 对象数组
- 指向对象的指针
- `this` 指针
- 对象的 `const` 指针
- 对象引用及对象的常引用
- 对象作为函数参数
- 嵌套类
- 局部类

3.4.2 教学目标

- 掌握对象数组的使用
- 理解 `this` 指针的含义及使用
- 掌握对象引用的使用
- 了解嵌套类及局部类的使用
- 掌握对象用做函数的参数和返回值

3.4.3 重点难点

- 静态成员的定义及应用
- `this` 指针的使用
- 函数重载技术

3.5 继承

3.5.1 主要内容

- 继承与派生的基本概念
- 基类与派生类之间的关系
- 派生类的定义
- 继承方式
- 单重继承
- 多重继承
- 基类和派生类间的赋值兼容规则
- 同名隐藏

3.5.2 教学目标

- 理解继承的概念
- 掌握如何使用继承
- 理解三种继承方式
- 掌握单重继承及多重继承中构造函数的调用
- 掌握单重继承及多重继承中析构函数的调用
- 了解基类和派生类间的赋值兼容规则
- 理解同名隐藏

3.5.3 重点难点

- 派生类的构造与析构以及数据的初始化和调用顺序
- 派生类的定义，派生类的生成过程

3.6 运算符重载

3.6.1 主要内容

- 运算符重载简介及语法
- 用成员函数重载运算符
- 几种常用运算符的重载
- 不同类型数据间的转换

3.6.2 教学目标

- 掌握 C++ 中运算符重载的含义
- 掌握 C++ 中常见的运算符的重载形式
- 了解 C++ 中类类型转换的几种手段

3.6.3 重点难点

- 运算符的重载

3.7 多态

3.7.1 主要内容

- 多态的引述
- 多态的核心虚函数
- 隐藏、覆盖、重载
- 纯虚函数与抽象类
- 接口类
- 动态多态

3.7.2 教学目标

- 掌握 C++ 中虚函数的概念
- 理解 C++ 中动态多态
- 理解重载、隐藏、覆盖
- 了解抽象类的概念和作用
- 掌握纯虚函数的定义和使用

3.7.3 重点难点

- 动态多态
- 重载、隐藏、覆盖
- 纯虚函数的定义和使用

3.8 动态数组（线性表的顺序存储）

3.8.1 主要内容

- 动态数组结构的定义
- 动态数组的创建
- 动态数组数据插入
- 动态数组数据删除
- 动态数组数据查找
- 动态数组合并
- 动态数组的销毁

3.8.2 教学目标

- 掌握动态数组的结构声明
- 掌握动态数组的常见算法

3.8.3 重点难点

- 动态数组的结构声明
- 动态数组的常见算法

3.9 链表（线性表的链式存储）

3.9.1 主要内容

- 链表结构的定义
- 链表的创建
- 链表数据插入
- 链表数据删除
- 链表数据查找
- 链表合并
- 链表的销毁

3.9.2 教学目标

- 掌握链表的结构声明
- 掌握链表的常见算法

3.9.3 重点难点

- 链表的结构声明
- 链表的常见算法

3.10 二叉树（二叉树的二叉链表存储）

3.10.1 主要内容

- 二叉树结构的定义
- 二叉树的创建
- 二叉树的先序遍历
- 二叉树的中序遍历
- 二叉树的后序遍历
- 二叉树的销毁

3.10.2 教学目标

- 掌握二叉树的结构声明
- 掌握二叉树的常见算法

3.10.3 重点难点

- 二叉树的结构声明
- 二叉树的常见算法

第四部分 教学相关资料

教材

- ✓ 面向对象程序设计及 C++（第 2 版）； 朱立华，俞琮 著； 人民邮电出版社。

教学参考书及网站

- ✓ C++编程思想（两卷合订本）； [美] Bruce Eckel, Chuck Allison 著； 刘宗田 等 译； 机械工业出版社。
- ✓ C++ Primer（第 5 版）； [美] Stanley B. Lippman, Josée Lajoie, Barbara E. Moo 著； 王刚，杨巨峰 译； 电子工业出版社。

执 笔 人：丁盟

审 定 人：

批 准 人：教学研究与管理委员会