**《C++程序设计》课程教学大纲**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 英文课程名 | Object Oriented Programming | 总 学 时 | | 64 | 学 分 | 4 |
| 课程编码 | G126201 | 理论教学学时 | | 48 | 线上教学学时\* | 0 |
| 开课学院（部） | 计算机科学与技术学院 | 实践  教学  学时 | 实验学时 | 16 | 先修课程 | 先修《程序设计基础C》  并修《离散数学》 |
| 课程类别 | ☑ 大类基础课程 □ 专业课程 | 上机学时 | 16 | 适用专业 | 数据科学与大数据技术专业 |
| ☑ 理论课程 □ 实践课程 | 其它 | 0 | 基层教学组织 | 计算机基础课程群教学团队 |
| ☑ 必修 □ 选修 | 开课平台 |  | | 课程链接 |  |
| 教学类型\* | | ☑ 线下教学 □ 线上线下混合式教学 □ 线上教学 □ 双语 □ 全英语 | | | | |

备注：采用线上教学、线上线下混合式教学需经所在学院同意并报教务处审核批准。

# 一、课程简介

C++是一种支持面向对象程序设计的语言, 语言简洁、紧凑，层次清晰，表现能力和处理能力极强。它具有丰富的运算符和数据类型，便于实现各类复杂的数据结构。C++既可用于系统软件的开发，也适合于应用软件的开发。本课程在学习C++的基本语法、编程特点的基础上，再进一步学习面向对象的C++，就能以一种熟悉的语法来学习面向对象的编程思想。因此，C++是计算机专业的重要必修基础课程。

# 二、课程教学目标

课程教学目标1：理解并掌握包括类定义、类继承、类封装以及多态性，继承性等面向对象的基本概念以及相关的编程方法。能够运用这些语法规则设计应用需要的类以及将类运用到具体的应用实际中。掌握编制基于类的具有一定复杂性的程序，并具备使用这些知识求解一定的程度的复杂应用问题。

课程教学目标2：理解并掌握C++的语言的面向对象编程思想、使学生深刻理解并掌握面向对象的程序设计语法规则和编程方法。能够认识到与面向过程的程序设计方法之间的差异。全面掌握C++程序设计语言。

课程教学目标3：深入掌握C++的一种软件开发环境(如 visual C++)，掌握面向对象程序开发的相关的编程技术、测试调试方法。在前面的程序设计语言基础的学习前提下进一步提高实践操作能力。强化培养学生利用开发工具解决/调试工程软件开发的能力。

# 三、课程教学目标与毕业要求对应关系

本课程支撑以下毕业要求指标点：

指标点1-3：能够将工程基础和专业知识用于求解数据科学领域复杂工程问题。

指标点3-1：掌握解决复杂数据科学与技术问题相关开发语言。

指标点5-1：了解现代工程工具和信息技术工具的使用方法，以及实际数据分析、数据工程实践中现代工具的使用现状。

本课程教学目标对毕业要求的支撑关系及其权重如表1所示。

表1 课程教学目标对毕业要求指标点的支撑关系及其权重

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 课程教学目标 | 毕业要求1-3 | 毕业要求3-1 | 毕业要求5-1 |
| 1 | 课程教学目标1 | 1 |  |  |
| 2 | 课程教学目标2 |  | 1 |  |
| 3 | 课程教学目标3 |  |  | 1 |
| 合计 | | 1 | 1 | 1 |

# 四、课程教学内容与学时分配

## 1. 理论课教学安排

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 章节或知识点 | 教学内容 | 教学重点、难点，课程思政要素 | 学时  分配 | 教学要求 | 教学方式 | 学生任务 | | 所支撑  课程目标 |
| 作业要求 | 其他要求(自学/讨论） |
| 1 | 回顾C++过程化编程知识 | 复习回顾C++过程化编程知识，包括: 基本类型和表达式、控制结构基本语法、函数、数组、指针、结构等。 | 教学重点：  复习过程化编程思想；巩固流程图的概念  教学难点：   1. 讲授C++面向过程化编程涉及的相关语法： 数据基本类型、控制结构、函数、复合类型等。特别对结构这一复合类型做重点回顾。 2. 通过具体的函数举例回顾函数功能和流程描述的工具：流程图。   课程思政要素：  社会责任：树立为我国软件产业发展做贡献的意识。 | 2 | 1. 培养学生对所学知识温故知新的能力； 2. 复习过程化编程的思想。掌握基本类型；三种基本的程序控制结构：顺序、选择、循环；函数声明及定义的语法以及如何提炼问题的功能模块设计对应的函数。 3. 复习程序流程图、盒图等方法规范，帮助学生养成使用图形来描述复杂问题的习惯。 4. 具有为国产软件自主研发和应用推广贡献力量的觉悟和意识，具有社会责任感和家国情怀。 | 讲授+演示+例题讲解 | 复习巩固结构化编程相关的练习；  复习巩固流程图相关的练习； | 讨论：  你对腾讯公司被制裁事件怎么看？你对我国民族产业的发展有什么想法？ | 课程目标2 |
| 2 | 指针的复杂应用 | 单链表节点（struct node）定义；  链表的基本操作和应用：链表的创建、链表访问、删除、查找；  复杂操作应用：逆置、合并、排序等。 | 教学重点：   1. 链表的基本定义方法和基本操作应用； 2. 能掌握基本的链表操作：链表的建立、访问、删除、查找等。 3. 理解链表与数组在数据存储组织上的差异。   教学难点：  掌握基本的链表操作：链表的建立、访问、删除、查找等。  课程思政要素：  社会责任：树立为我国软件产业发展做贡献的意识。 | 4 | 1. 掌握链表结点的定义方法，理解指向自己类型的指针即链指针； 2. 能掌握基本的链表操作：链表的建立、访问、删除、查找等。 3. 选择掌握链表的复杂操作应用； 4. 理解链表与数组在数据存储组织上的差异。 5. 具有为国产软件自主研发和应用推广贡献力量的觉悟和意识，具有社会责任感和家国情怀。 | 讲授+演示+例题讲解 | 链表基础概念、链表的操作、链表的应用相关的练习； | 讨论：  链表操作的关键核心以及链表的变形？ | 课程目标2，  课程目标3 |
| 3 | 类的  定义 | 从结构（struct）过渡到类的引入;  类的声明和定义;  面向对象开发中程序的组织;  类的成员函数和数据成员;  类的访问控制。 | 教学重点：   1. 类的基本语法； 2. 学会定义和使用类，使用类的成员函数；理解封装性。   教学难点：   1. 类的基本访问控制，理解各种访问控制的差异； 2. 学会定义和使用类，使用类的成员函数；理解封装性，会在实践中定义和组织包含类应用的程序。   课程思政要素：  工匠精神：培养编程时一丝不苟、严谨精准的工匠精神。 | 4 | 1. 掌握class的基本语法：声明和定义一个类； 2. 掌握类内的两类成员定义的语法：成员函数和数据成员； 3. 掌握成员函数在类内实现和类外实现的语法差异； 4. 掌握基于类的面向对象程序开发的组织结构。 5. 掌握类的基本访问控制。 6. 学会定义和使用类，使用类的成员函数；理解封装性。 7. 能够在编程过程中形成一丝不苟、精准控制的严谨意识和工匠精神。 | 教学方法：讲授+演示+例题讲解 | 类的基本概念、类的基本访问控制相关的练习； | 讨论：  类的语法以及类内的访问控制方式 | 课程目标1，  课程目标2，  课程目标3 |
| 4 | 类与  对象 | 构造与析构（构造函数与析构函数）；  带默认参数的函数；  函数重载；  对象构造的顺序；  类的数据成员初始化；  拷贝构造函数及应用场合；  const和static在类中的使用（常函数；常数据成员；静态成员）  对象指针，包括this指针；  类的引用类型；  赋值运算符成员函数；  友元；  操作符重载；  ***赋值与转型*** 。 | 教学重点：   1. 理解类内的四类可缺省成员函数的功能和局限性（无参构造、拷贝构造、赋值重载和析构函数） 2. 理解和掌握Ｃ＋＋各类运算符在类中重载的方法（类内和类外）； 3. 系统掌握构造和析构的顺序，数据成员的初始化顺序；   教学难点：   1. 理解和掌握Ｃ＋＋各类运算符在类中重载的方法（类内和类外）； 2. 理解const和static在类内的使用情况，对成员函数和数据成员的约束； 3. 理解和掌握Ｃ＋＋各类运算符在类中重载的方法（类内和类外）； 4. friend的使用； 5. 对象指针，类的引用，赋值及转型的概念。   课程思政要素：  工匠精神：培养编程时一丝不苟、严谨精准的工匠精神。 | 14 | 1. 理解类的结构； 2. 掌握类的成员函数和析构函数定义的语法； 3. 理解类内的四类可缺省成员函数的功能和局限性（无参构造、拷贝构造、赋值重载和析构函数） 4. 理解const和static在类内的使用情况，对成员函数和数据成员的约束； 5. 了解友元的含义以及使用场合； 6. 理解和掌握Ｃ＋＋各类运算符在类中重载的方法（类内和类外）； 7. 系统掌握构造和析构的顺序，数据成员的初始化顺序； 8. 系统掌握对象指针，类的引用，赋值及转型的概念。 9. 能够在编程过程中形成一丝不苟、精准控制的严谨意识和工匠精神。 | 讲授+演示+例题讲解 | 类的四类可缺省成员以及类的构造、析构次序理解相关的练习；  类的友元的概念以及友元的使用相关的练习；  const、statci成员的理解相关的练习；  类的运算符重载相关的练习； | 讨论：  理解4种可缺省的类内成员的特点？什么时候需要自定义？ | 课程目标1，  课程目标2，  课程目标3 |
| 5 | 类的  继承 | 类的继承概念、层次结构；  继承访问控制、访问父类成员；  派生类的构造；  继承与组合。 | 教学重点：   1. 结合实例讲授public, protected, 2. private三类访问控制，并结合类内访问控制介绍访问控制对类成员的约束，以及在继承模型中的作用；特别强调protected在继承中的作用； 3. 能结合类内成员的访问控制理解和掌握子类对父类成员的访问； 4. 理解和掌握在继承模型下构造和析构的次序；Ｃ＋＋中的继承模型；   教学难点：   1. 理解和掌握继承分层、派生现象，进而理解派生类的构造和继承方法。 2. 理解和掌握三类继承访问控制; 3. 能结合类内成员的访问控制理解和掌握子类对父类成员的访问； 4. 理解和掌握在继承模型下构造和析构的次序；   课程思政要素：  工匠精神：培养编程时一丝不苟、严谨精准的工匠精神。 | 6 | 1. 理解和掌握继承分层、派生现象，进而理解派生类的构造和继承方法。 2. 理解和掌握三类继承访问控制; 3. 能结合类内成员的访问控制理解和掌握子类对父类成员的访问； 4. 理解和掌握在继承模型下构造和析构的次序； 5. 能在已经定义的类的基础上定义新类－继承和组合的使用； 6. 能够在编程过程中形成一丝不苟、精准控制的严谨意识和工匠精神。 | 讲授+演示+例题讲解 | 类的继承、三类继承访问控制的理解相关的练习；  能结合类内成员的访问控制理解和掌握子类对父类成员的访问相关的练习；  在继承模型下类的构造析构的次序理解的相关练习； | 讨论：  继承和组合有什么不一样？ | 课程目标1，  课程目标2，  课程目标3 |
| 6 | 类的多态性 | 类的多态性概念  类的抽象  虚函数  ***多态编程*** | 教学重点：   1. 多态的概念； 2. 虚函数的语法及作用   教学难点：   1. 结合实例演示并讲解不同事物同名操作下的不同实现。 2. 讲授Ｃ＋＋中面向对象的多态表现为静态多态和动态多态。静态主要是函数重载；动态则使用virtual关键字。 3. 帮助学生了解影响抽象编程的因素。培养快速全盘把握问题的抽象化方法与手段，不为细节所累的能力。   课程思政要素：  工匠精神：培养编程时一丝不苟、严谨精准的工匠精神。 | 6 | 1. 理解和掌握面向对象中多态的概念； 2. 理解和掌握虚函数的语法及作用； 3. 理解和掌握静态多态；进一步掌握动态； 4. 能够在编程过程中形成一丝不苟、精准控制的严谨意识和工匠精神。 | 讲授+演示+例题讲解 | 多态相关的练习；  虚函数相关的练习； | 讨论：  静态多态与动态多态的比较？抽象类可以实例化相应的对象吗？ | 课程目标1，  课程目标2，  课程目标3 |
| 7 | 抽象  方法 | 抽象基类  虚基类与纯虚函数 | 教学重点：   1. 结合实例演示并讲解虚基类的定义及使用方法； 2. 进一步讲授纯虚函数的定义语法和含义，以及应用茶馆和； 3. 讲授抽象类做界面或接口与具体类的联系和约束、具体类与应用程序的使用配合； 4. 讲授虚基类配合基类指针实现动态多态的类层次结构以及应用编程。培养学生问题抽象的能力，进一步全面建立面向对象程序设计的思想；理解面向对象程序设计中的类和对象的概念，以及如何将通过语言工具将现实世界中的事物建模到计算机中；   教学难点：   1. 理解和掌握虚基类和纯需基类的定义方法和差异； 2. 能结合实际的问题，建立对应的类模型；能够运用虚基类和基类指针构建动态多态模型，能用抽象类做接口，培养学生综合看问题的能力。   课程思政要素：  工匠精神：培养编程时一丝不苟、严谨精准的工匠精神。 | 6 | 1. 理解和掌握抽象类的定义语法，注意抽象类和具体类的关系及差异； 2. 理解和掌握虚基类和纯需基类的定义方法和差异； 3. 能结合实际的问题，建立对应的类模型；能够运用虚基类和基类指针/引用构建动态多态模型，能用抽象类做接口，培养学生综合看问题的能力； 4. 能够在编程过程中形成一丝不苟、精准控制的严谨意识和工匠精神。 | 讲授+演示+例题讲解 | 抽象类（纯虚函数），虚基类相关的练习；  利用虚基类及基类的指针、引用实现动态多态的练习； | 讨论：  抽象基类是否可以实例化？ 纯虚函数有什么特点？ | 课程目标1，  课程目标2，  课程目标3 |
| 8 | ***模板与异常处理*** | ***函数模板与模板类***  ***异常处理*** | 教学重点：   1. 理解和掌握函数模版、类模版； 2. 理解和掌握Ｃ＋＋异常处理的概念和相应的关键字； 3. 掌握异常处理的通用模型对应的语法，能简单应用异常处理。   教学难点：   1. 讲授template关键字在普通函数和类定义中的使用语法，结合实例演示模版函数与函数重载的差异；演示模版类的实际使用方法，特别是模版类内的函数成员的定义实现的注意点； 2. 讲授try, throw, catch这三个关键字在异常处理中的使用语法和作用，介绍Ｃ＋＋异常处理的一些规则，结合例题演示包含异常处理的程序。   课程思政要素：  工匠精神：培养编程时一丝不苟、严谨精准的工匠精神。 | 6 | 1. 理解和掌握函数模版、类模版； 2. 理解和掌握涉及模版的工程文件的组织； 3. 理解和掌握Ｃ＋＋异常处理的概念和相应的关键字； 4. 掌握异常处理的通用模型对应的语法，能简单应用异常处理； 5. 能够在编程过程中形成一丝不苟、精准控制的严谨意识和工匠精神。 | 讲授+演示+例题讲解 | 函数模版、类模版定义和使用的练习；  C++基本的异常处理的概念和简单应用相关的练习； | 讨论：  C++中的template关键字的使用？C++中的异常处理机制涉及的关键字以及使用方法？ | 课程目标1，  课程目标2，  课程目标3 |

注：***其中黑斜体部分内容为选学内容，授课教师可根据具体教学安排酌情讲解或组织学生自学。***

## 2. 实践教学安排

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 项 目 | 学时或周数 | 类型 | 每组人数 | 教学要求 | 教学方式 | 学生任务 | 所支撑  课程目标 |
| 1 | 单链表创建及其他操作 | 2 | 设计 | 1 | 1. 掌握单链表节点定义及链表创建，完成链表的基本操作，如访问，查找等； 2. 进行程序设计、编程、调试、运行和结果分析。 | 讲授+任务驱动式教学+上机实践 | 1. 实验前提前预习实验内容。 2. 设计程序框架。 3. 编码实现。 4. 不断精益求精，改进设计，最终实现实验目标。 5. 实验后进行总结与反思，撰写实验报告并按时提交。 | 课程目标1，  课程目标2，  课程目标3 |
| 2 | 类与对象1：类的声明、定义和使用 | 2 | 设计 | 1 | 1. 要求掌握类的定义、关键字使用以及对.h和.cpp文件的区分。掌握含有自定义类库的工程文件的组织； 2. 掌握类的使用方法，对象的实例化声明以及类的成员函数的使用； 3. 进行程序设计、编程、调试、运行和结果分析。 | 讲授+任务驱动式教学+上机实践 | 1. 实验前提前预习实验内容。 2. 设计程序框架。 3. 编码实现。 4. 不断精益求精，改进设计，最终实现实验目标。 5. 实验后进行总结与反思，撰写实验报告并按时提交。 | 课程目标1，  课程目标2，  课程目标3 |
| 3 | 类与对象2：类的构造、拷贝构造函数、赋值运算符重载和析构 | 2 | 设计 | 1 | 1. 能正确理解缺省类成员函数的功能，知道在什么样的场合需要自己手工重写可缺省部分； 2. 能验证构造和析构的次序； 3. 能正确使用赋值运算重载（类内成员）的形式； 4. 进行程序设计、编程、调试、运行和结果分析。 | 讲授+任务驱动式教学+上机实践 | 1. 实验前提前预习实验内容。 2. 设计程序框架。 3. 编码实现。 4. 不断精益求精，改进设计，最终实现实验目标。 5. 实验后进行总结与反思，撰写实验报告并按时提交。 | 课程目标1，  课程目标2，  课程目标3 |
| 4 | 类与对象3：类的操作符重载 | 2 | 设计 | 1 | 1. 掌握类的操作符重载，能够编程实现操作符重载； 2. 能够区分哪些运算符必须在类内重载，而有些运算符既可以在类内也可以在类外重载，而有些Ｃ＋＋的运算符不允许重载； 3. 能够掌握在类内重载和在类外作为普通函数重载的语法； 4. 进行程序设计、编程、调试、运行和结果分析。 | 讲授+任务驱动式教学+上机实践 | 1. 实验前提前预习实验内容。 2. 设计程序框架。 3. 编码实现。 4. 不断精益求精，改进设计，最终实现实验目标。 5. 实验后进行总结与反思，撰写实验报告并按时提交。 | 课程目标1，  课程目标2，  课程目标3 |
| 5 | 类的继承1：类的继承及派生类使用 | 2 | 设计 | 1 | 1. 掌握类的派生、继承，能够编程实现派生类，并对派生类中的方法进行正确调用。 2. 进行程序设计、编程、调试、运行和结果分析。 | 讲授+任务驱动式教学+上机实践 | 1. 实验前提前预习实验内容。 2. 设计程序框架。 3. 编码实现。 4. 不断精益求精，改进设计，最终实现实验目标。 5. 实验后进行总结与反思，撰写实验报告并按时提交。 | 课程目标1，  课程目标2，  课程目标3 |
| 6 | 类的多态性1：虚函数的使用 | 2 | 设计 | 1 | 1. 掌握虚函数的定义和使用，对虚函数在派生类中的使用能够正确理解并编程实现 2. 能够根据问题模型自行设计简单的继承层次用于求解； 3. 进行程序设计、编程、调试、运行和结果分析。 | 讲授+任务驱动式教学+上机实践 | 1. 实验前提前预习实验内容。 2. 设计程序框架。 3. 编码实现。 4. 不断精益求精，改进设计，最终实现实验目标。 5. 实验后进行总结与反思，撰写实验报告并按时提交。 | 课程目标1，  课程目标2，  课程目标3 |
| 7 | 类的多态性 | 2 | 设计 | 1 | 1. 掌握静态多态和动态多态的概念； 2. 能够使用虚函数以及指向基类的指针实现动态绑定的应用； 3. 进行程序设计、编程、调试、运行和结果分析。 | 讲授+任务驱动式教学+上机实践 | 1. 实验前提前预习实验内容。 2. 设计程序框架。 3. 编码实现。 4. 不断精益求精，改进设计，最终实现实验目标。 5. 实验后进行总结与反思，撰写实验报告并按时提交。 | 课程目标1，  课程目标2，  课程目标3 |
| 8 | **一个复杂的类定义及实现（*类模版）*** | 2 | 设计 | 1 | 1. 能够通过设计一个类实现较复杂的数据类型，并通过编程实现这个类；可以应用模版； 2. 掌握一定的解决复杂工程问题的程序设计语言基础； 3. 可以完善程序，使用基本的异常处理； 4. 进行程序设计、编程、调试、运行和结果分析。 | 讲授+任务驱动式教学+上机实践 | 1. 实验前提前预习实验内容。 2. 设计程序框架。 3. 编码实现。 4. 不断精益求精，改进设计，最终实现实验目标。 5. 实验后进行总结与反思，撰写实验报告并按时提交。 | 课程目标1，  课程目标2，  课程目标3 |

注：教师可根据实际的教学情况适当调整实践教学内容，题目数量可适当增减，实验学时也可调整。

# 五、考核内容、考核方式及评分标准

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 考核环节 | 考核方式 | 评分标准 | | | | | 占总成绩比例 |
| 90-100 | 80-90 | 70-80 | 60-70 | 60以下 |
| 平时作业 | / | 按时上交，次数齐全，书写整洁，对知识点掌握扎实，答案平均正确率90%（含）以上。 | 按时上交，次数齐全，书写整洁，对知识点掌握比较扎实，个别知识点有漏洞，答案平均正确率80%（含）~90%。 | 延迟上交或缺次不超过2次，书写一般，部分知识点掌握不牢固，答案平均正确率70%（含）~80%。 | 延迟上交或缺次3次，书写一般，部分概念混淆，知识点理解不透彻，答案平均正确率60%（含）~70%。 | 延迟上交或缺次超过3次，书写乱，概念模糊，知识点不理解，答案平均正确率60%以下。 | 20% |
| 实验 | 课程设置七到八次课内实验，每次实验前需要预习、设计程序框架、编码实现和调试。 | 全部实验在规定时间内完成并提交实验报告，90%（含）以上的功能实现，实验代码质量高，运行结果流程且实验报告完善。 | 全部在规定时间内完成并提交实验报告，且80%（含）~90%的功能实现，实验代码质量较高，运行结果正确且实验报告较完善。 | 延迟完成或延迟提交实验报告，70%（含）~80%的功能实现，实验代码完整，有一定的注释，可以运行，有小瑕疵且实验报告各部分完整。 | 缺席1次实验以下，60%（含）~70%的功能实现，实验代码完整，但运行有一定的问题，运行结果符合最低要求，实验报告各部分完整。 | 缺席2次及以上实验，60%（含）~70%的功能实现，超过40%的功能未实现，  实验代码不完整，实验报告不完整。 | 20% |
| 其他（自学/讨论） | 其他形式包括课堂提问、小组讨论、课堂测验等。 | 1、课堂测验成绩90-100；  2、积极参加课堂提问及讨论，且回答完全正确，有独到见解。 | 1、课堂测验成绩80-90；  2、参加课堂提问及讨论次数70%以上，且回答基本正确。 | 1、课堂测验成绩70-80；  2、参加课堂提问及讨论次数50%以上，且回答基本正确。 | 1、课堂测验成绩60-70；  2、参加课堂提问及讨论1次以上。 | 1、课堂测验成绩60以下；  2、未参加课堂提问及讨论。 | 10% |
| 期末考试 | 采用闭卷考试 | 按试卷评分标准执行 | | | | | 50% |

# 五、教材及参考书目

教 材：[1] 谭浩强.《C++程序设计》（第3版）.清华大学出版社,2015,8. ISBN: 9787302408307 （大类教材）

参考书：

[1] 钱能. 《C++程序设计教程（第二版）》. 清华大学出版社，2005.

[2] 钱能. 《C++程序设计教程（第二版）习题与解答》. 清华大学出版社，2009.7.

[3] [[美]普拉塔(Prata,S.)](http://www.douban.com/book/search/%5B%E7%BE%8E%5D%E6%99%AE%E6%8B%89%E5%A1%94%EF%BC%88Prata%2CS.%EF%BC%89)，[孙建春](http://www.douban.com/book/search/%E5%AD%99%E5%BB%BA%E6%98%A5)/[韦强](http://www.douban.com/book/search/%E9%9F%A6%E5%BC%BA)译. 《C++Primer Plus（第五版）中文版》. 人民邮电出版社，2005.1.

[4] 侯杰，孟岩译. 《C++标准程序库》. 华中科技大学出版社，2002.9.

[5] Stanley B.Lippman，Josee Lajoie．，王刚等译. 《C++ Primer》（第五版）. 电子工业出版社，2013.

[6] Bruce Eckel著. 《C++编程思想（第一卷）》. 机械工业出版社，2002.9.

**执笔者：姜娓娓**

**审核者：杨良怀**

**课程教学团队成员：杨良怀、蒋莉、毛国红、钱能、田贤忠、王英姿、王海霞、姜娓娓**