

《面向对象程序设计（C++）》课程教学大纲

课程英文名	Object Oriented Programming (C++)				
课程代码	A0502380	课程类别	学科基础课	课程性质	学科必修
学分	3		总学时数	48	
开课学院	计算机学院		开课基层教学组织	程序设计课程组	
面向专业	计算机科学与技术（含第二学士学位）、软件工程、计算机科学与技术(计算机科学英才班)、智能计算与数据科学(计算机科学与技术)		开课学期	第 2 学期	

一、课程目标

《面向对象程序设计(C++)》是计算机类相关专业的一门重要的学科基础课，也是学习后续课程的重要基础。课程主要以 C++语言为依托，主要传授面向对象程序设计思想，讲授 C++程序设计方法，使学生初步具备根据实际问题，设计结构良好、高效的现代 C++程序的能力。结合国家建设和民族复兴的新时代背景，增强学生家国情怀与文化自信，激发学生使命感和责任心。

通过本课程的学习，要达到以下课程目标：

课程目标 1：能够运用 C++语言的各种机制，具备阅读和调试 C++程序的基本能力。

课程目标 2：具备运用面向对象思维方式，为实际应用问题建立问题模型，设计合适 C++类的初步能力。

课程目标 3：能够综合运用 C++语言机制，设计结构良好、高效的现代 C++程序。

课程目标 4：能够掌握面向对象程序设计的新技术与发展趋势，及时掌握国家相关方面的科技战略需求，树立强烈的爱国主义使命感与责任心。

二、课程目标与毕业要求对应关系

本课程的课程目标对计算机科学与技术专业毕业要求指标点的支撑情况如表 1 所示。

表 1. 课程目标与计算机科学与技术专业（含第二学士学位）毕业要求对应关系

毕业要求	指标点	课程目标及支撑权重
------	-----	-----------

1.工程知识：掌握数学、自然科学、工程基础、计算机专业领域的知识，并能应用于计算机领域复杂工程问题的解决方案中。	1-2 掌握计算机科学核心知识与理论，能够针对计算机领域复杂工程问题建立模型，并利用模型解决问题。	目标 1： 0.5 目标 2： 0.5
2.问题分析：能够应用数学、自然科学和工程科学的基本原理，以及科学思维方法，对计算机领域的复杂工程问题进行识别、表达和分析，并通过文献查阅与研究获得有效结论。	2-1 能够应用数学、自然科学、工程科学和计算科学的基本原理识别、表达计算机领域的复杂工程问题。	目标 2： 1.0
3.设计/开发解决方案：能够设计计算机领域复杂工程问题的解决方案，设计与开发满足特定需求的软硬件系统、算法或部件，在设计中考虑社会、健康、安全、法律、文化以及环境等因素，并体现一定的创新意识。	3-1 具备计算思维和程序设计能力，能够针对计算机复杂系统设计与开发满足特定需求的模块或算法。	目标 2： 0.5 目标 3： 0.5
12.终身学习：具有自主学习和终身学习的意识，有不断学习和适应信息技术高速发展的能力。	12-2 具备自主学习的能力，包括技术理解力、归纳总结能力和提出问题的能力。	目标 4： 1.0

本课程的课程目标对软件工程专业毕业要求指标点的支撑情况如表 2 所示。

表 2. 课程目标与软件工程专业毕业要求对应关系

毕业要求	指标点	课程目标及支撑权重
1. 工程知识：掌握数学、自然科学、工程基础、软件工程专业领域的知识，并能应用于软件工程专业领域复杂工程问题的解决方案中。	1-2 掌握软件工程核心知识与理论，能够针对软件工程领域复杂工程问题建立模型，并利用模型解决问题。	目标 1： 0.5 目标 2： 0.5
2. 问题分析：能够应用数学、自然科学、工程科学的基本原理和科学思维方法，对软件工程领域复杂工程问题进行识别、表达和分析，并通过文献查阅与研究获得有效结论。	2-1 能够应用数学、自然科学、工程科学和软件工程的基本原理识别、表达软件工程领域复杂工程问题。	目标 2： 1.0
3. 设计/开发解决方案:能够设计软件工程领域复杂工程问题的解决方案，设计与开发满足特定需求的软件系统、模块或算法，在设计中考虑社会、健康、安全、法律、文化以及环境等因素，并体现一定的创新意识。	3-1 具备计算思维和程序设计能力，能够针对复杂软件系统设计与开发满足特定需求的模块或算法。	目标 2： 0.5 目标 3： 0.5
12.终身学习：具有自主学习和终身学习的意识，有不断学习和适应信息技术高速发展的能力。	12-2 具备自主学习的能力，包括技术理解力、归纳总结能力和提出问题的能力。	目标 4： 1.0

本课程的课程目标对智能计算与数据科学（计算机科学与技术）专业毕业要求指标点的支撑情况如表 3 所示。

表 3. 课程目标与智能计算与数据科学（计算机科学与技术）专业毕业要求对应关系

毕业要求	指标点	课程目标及支撑权重
------	-----	-----------

1.工程知识：能够掌握数学、自然科学、工程基础、计算机软硬件知识、人工智能、智能计算和数据科学的基础理论及专业知识，并应用在人工智能、智能计算和大数据专业领域及其他相关交叉领域的复杂工程问题的解决方案中。	1-2 掌握计算机科学核心知识与理论，能够针对计算机领域复杂工程问题建立模型，并利用模型解决问题。	目标 1： 0.5 目标 2： 0.5
2.问题分析：能够应用数学、自然科学、工程科学、人工智能、智能计算和大数据的基本原理，对人工智能和大数据专业领域及其他相关交叉领域的复杂工程问题进行识别、表达、分析和抽象建模，并通过文献查阅与研究获得有效结论。	2-1 能够应用数学、自然科学、工程科学和计算科学的基本原理识别、表达计算机领域的复杂工程问题。	目标 2： 1.0
3.设计/开发解决方案：能够设计人工智能、智能计算和大数据专业领域及其他相关交叉领域复杂工程问题的解决方案，能够设计与开发满足特定需求的计算机软硬件系统、模型或算法，在设计中考虑社会、健康、安全、法律、文化以及环境等因素，并体现一定的创新意识。	3-1 具备计算思维和程序设计能力，能够针对计算机复杂系统设计与开发满足特定需求的模块或算法。	目标 2： 0.5 目标 3： 0.5
12.终身学习：具有自主学习和终身学习的意识，有不断学习和适应信息技术高速发展的能力。	12-2 具备自主学习的能力，包括技术理解力、归纳总结能力和提出问题的能力。	目标 4： 1.0

本课程的课程目标对计算机科学英才班（计算机科学与技术）专业毕业要求指标点的支撑情况如表 4 所示。

表 4. 课程目标与计算机科学英才班（计算机科学与技术）专业毕业要求对应关系

毕业要求	指标点	课程目标及支撑权重
1.工程与科学知识：能够掌握数学、自然科学、工程基础、计算机科学理论知识，并应用在计算机相关领域的复杂工程问题和基础科学问题的解决方案中。	1-2 掌握计算机科学核心知识与理论，能够针对计算机复杂工程问题建立模型，并利用模型解决问题。	目标 1： 0.5 目标 2： 0.5
2.问题分析：能够应用数学、自然科学和工程科学的基本原理，以及计算科学思维方法，对计算机相关领域的复杂工程问题进行抽象分析与识别、建模表达和形式化论证，并通过文献查阅与研究获得有效结论。	2-1 能够应用数学、自然科学、工程科学和计算科学的基本原理识别、表达计算机相关领域的复杂工程问题。	目标 2： 1.0
3.设计/开发解决方案：能够设计计算机相关领域复杂工程问题的解决方案，能够设计与开发满足特定需求的计算机软硬件系统、模块或算法，在设计中考虑社会、健康、安全、法律、文化以及环境等因素，并体现较强的创新意识，具备基本的创新能力。	3-1 具备计算思维和程序设计能力，能够针对计算机复杂系统设计与开发满足特定需求的模块或算法。	目标 2： 0.5 目标 3： 0.5
12.终身学习：具有较强的自主学习和终身学习的意识，有不断学习和适应信息技术高速发展的能力。	12-2 具备较强的自主学习的能力，包括技术理解力、归纳总结能力和提出问题的能力。	目标 4： 1.0

三、 课程目标与教学内容和方法的对应关系

课程目标与教学内容、建议教学方法的对应关系如表 5 所示。教师可根据实际情况和自身特长采用合适的教学方法。

表 5 课程目标与教学内容、建议教学方法的对应关系

教学内容	建议教学方法	课程目标
1.C++程序设计基础	课堂讲授、案例分析、课后实践	1,4
2.类和对象	课堂讲授、案例分析、课后实践	1,2,3
3.拷贝控制	课堂讲授、案例分析、课后实践	1,2,3
4.运算符重载	课堂讲授、案例分析、课后实践	1,2,3,4
5.继承和派生	课堂讲授、案例分析、课后实践	1,2,3
6.模板	课堂讲授、案例分析、课后实践	1,2,3,4
7.异常和智能指针	课堂讲授、案例分析、课后实践	1,2,3,4
8.C++标准模板库简介	课堂讲授、案例分析、课后实践	1,2,3,4

课程教学的详细内容与要求如下：

1. C++程序设计基础

(1) 教学内容：

- C++简史，面向对象方法与面向过程方法；
- 函数重载、内联函数、引用；
- 栈与函数调用的实现；
- 作用域和生存期知识；
- 动态分配技术的典型案例；

(2) **教学重点：**函数重载，参数引用传递，作用域和生存期知识，动态分配技术。

(3) **教学难点：**动态分配技术的运用。

(4) **教学要求：**能够正确理解和使用函数重载、内联函数、引用参数传递，理解栈与函数调用的实现，掌握连续空间的动态分配和链表处理。

思政融合点 1：介绍 C++的应用领域和应用前景，使学生能够把自身价值的实现和国家发展紧密联系起来。介绍学习面向对象编程技术的必要性和重要性，使学生明白学好本课程能为接下来专业课的学习打好基础，从而更好地进行创新实践和科学研究。

思政融合点 2：C++是一种适合系统、工具软件开发的编程语言，结合国家相关方面的科技战略需求，帮助学生树立强烈的爱国主义使命感与责任心。

思政融合点 3：介绍 C++的应用领域和应用前景，使学生能够把自身价值的实现和国家发展紧密联系起来。介绍学习面向对象编程技术的必要性和重要性，使学生明白学好本课程能为接下来专

业课的学习打好基础，从而更好地进行创新实践和科学研究。

2. 类与对象

(1) 教学内容:

- 类和对象的概念;
- 简单类的设计和实现;
- C++常用容器的使用;
- 描述 has-a 关系的类类型数据成员
- 特殊场合使用的友元关系和静态成员;
- 典型案例-简单集合类的设计和实现;

(2) **教学重点:** 类和对象的概念, 简单类的设计和实现, C++常用容器的使用, 类类型数据成员, 友元关系和静态成员。

(3) **教学难点:** 类和对象概念的建立、简单类的设计和实现。

(4) **教学要求:** 能够建立类和对象的概念, 设计和实现简单类, 掌握 C++常用容器的简单使用, 正确使用类的组合, 掌握简单对象在函数间的传递, 理解简单的 UML 类图。

3. 拷贝控制

(1) 教学内容:

- 使用动态分配类对象的复制和赋值;
- 合成或实现拷贝控制的五个函数: 拷贝构造、移动构造、复制赋值、移动赋值、析构函数;
- 典型案例-链表表示的集合类实现;
- 链集合向量空间扩充问题;

(2) **教学重点:** 使用动态分配类对象的复制和赋值, 拷贝控制五个函数的自动合成或实现, 对象传递在函数间传递的效率问题。

(3) **教学难点:** 拷贝控制五个函数的合成或实现, 链集合向量空间扩充问题。

(4) **教学要求:** 能够掌握类对象的传递和赋值, 掌握类拷贝控制函数的缺省合成方法, 掌握必要时设计和实现使用动态分配类的五个拷贝控制函数, 理解复杂对象的内存状态。

4. 运算符重载

(1) 教学内容:

- 成员运算符重载;
- 友元运算符重载;
- 不同类型对象间的转换;
- 典型案例-字符串类的设计和实现;

(2) **教学重点:** 常用运算符的成员运算符、友元运算符方式重载, 不同类型对象间的转换, 字符串类的设计和实现。

(3) **教学难点:** 使用动态分配和运算符重载类的设计和实现。

(4) **教学要求:** 能够掌握常用运算符的成员运算符和友元运算符两种重载方式, 设计和实现

使用动态分配、运算符重载的类。

5. 继承和派生

(1) 教学内容:

- 公有继承和 is-a 关系;
- 赋值兼容原则、同名覆盖原则
- 动态多态性和虚函数;
- 典型案例-各类物体面积求和;
- 其它继承方式和多继承;
- 派生类对象的内存分布, 虚函数实现原理;
- 运行时类型转换;

(2) 教学重点: 派生类概念, 派生类成员的访问控制, 赋值兼容原则,

派生类对象的内存分布和构造、析构次序, 派生机制在提高软件可重用性方面的重要作用。

(3) 教学难点: 赋值兼容原则, 派生类对象的内存分布, 虚函数实现原理, 运行时类型转换。

(4) 教学要求: 能够掌握派生类概念、派生类成员的访问控制, 掌握派生类新成员定义方式和同名覆盖原则, 掌握赋值兼容原则, 掌握派生类对象的内存分布和构造、析构次序, 理解多继承、虚基类及可能的问题, 了解运行时类型转换。

6. 模板

(1) 教学内容:

- 函数模板;
- 类模板;
- 模板特化和偏特化;
- 典型案例-链栈类模板设计和实现。

(2) 教学重点: 典型类模板的设计和实现。

(3) 教学难点: 类、模板、动态分配的综合应用。

(4) 教学要求: 掌握 C++ 类模板设计和函数模板设计, 理解模板特化和偏特化, 掌握向量、栈、队列等典型容器类模板的设计和实现。

思政融合点 4: 本部分教学内容要求学生设计实现一个接口良好、功能完善的类模板。学生在学习和开发过程中, 不仅要编码实现, 还要搭建结构优良的软件架构, 通过这种实践增强学生的专业自信心, 从而树立研究前沿技术、不断创新的远大志向。

7. 异常和智能指针

(1) 教学内容:

- 异常抛出;
- 异常捕捉;
- 异常分类;
- 资源泄露和 RAII;

- 智能指针 `unique_ptr`、`shared_ptr`、`weak_ptr`;

- 异常安全性问题;

(2) **教学重点:** 异常抛出和捕捉机制, 智能指针 `unique_ptr`、`shared_ptr` 的使用。

(3) **教学难点:** 异常抛出和捕捉机制, 智能指针使用, 异常安全性问题。

(4) **教学要求:** 能够掌握异常抛出和捕捉机制, 学会异常分类方法, 学会现代 C++ 程序设计的 RAII 思想, 能够利用 C++ 11 智能指针 `unique_ptr` 或 `shared_ptr` 避免异常发生时的资源泄露, 了解异常安全性问题。

8. C++标准模板库简介

(1) **教学内容:**

- 标准模板库 (STL) 的容器、迭代器、函数对象和算法组件概念;
- 常用容器;
- 常用算法;
- lambda 表达式;

(2) **教学重点:** STL 常用容器和容器适配器, STL 常用算法, lambda 表达式。

(3) **教学难点:** STL 运作机制, lambda 表达式。

(4) **教学要求:** 能够了解泛型程序设计思想, 了解 STL 容器、迭代器、算法和函数对象的机制, 学会常用容器、容器适配器、常用算法和 lambda 表达式的使用。

思政融合点 5: 总结面向对象程序设计方法, 回顾面向对象领域的新技术与发展趋势, 结合国家相关方面的科技战略需求, 帮助学生树立强烈的爱国主义使命感与责任心。

四、 实践环节及要求

上机实践是本课程的重要环节之一, 通过上机实践, 使学生深入学习和掌握 C++ 程序的设计、运行和调试方法。表 6 列出了本课程需要完成的 8 个上机项目, 本课程课内上机 12 学时, 建议课内上机覆盖带*内容, 其它内容可安排在课外进行, 教师也可根据情况适当增删上机项目。

表 6 上机项目安排表

序号	上机项目	学时	基本要求	实验性质	实验类别
1	普通类设计和实现	3	完成普通类的设计、实现和应用	选做	综合
2	类的组合	3	完成组合关系类的设计、实现和应用	选做	综合
3	类和动态分配*	3	完成包含动态分配类的设计、实现, 掌握复杂对象的复制、赋值及在函数间的传递、转移	选做	综合
4	运算符重载 *	3	完成含运算符重载类的设计、实现和应用	选做	综合
5	继承和虚函数 *	3	设计具有继承关系的多个类, 体现动态多态性	选做	综合
6	类模板设计、实现 *	3	完成类模板的设计、实现及应用	选做	综合

7	异常和智能指针	3	掌握通过智能指针管理动态分配内存资源	选做	综合
8	STL 简单应用	3	掌握 STL 基本容器和迭代器的使用	选做	综合
合计		选 12			

五、 与其它课程的联系

先修课程：程序设计基础或程序设计与算法基础

后续课程：数据结构、操作系统、计算机网络

六、 学时分配

总学时 48 学时，其中课堂讲授、讨论及习题 36 学时，课内上机共 12 学时。任课教师可根据实际教学情况安排习题、讨论课时和上机内容，建议课时分配如表 7 所示。

表 7 学时分配表

教学内容	讲课时数	实验时数	实践学时	上机时数	自学时数	习题课	讨论时数
1. C++程序设计基础	6				2		
2. 类与对象	9				3		
3. 拷贝控制	3			3	2		
4. 运算符重载	4			3	2		
5. 继承和派生	3			3	2		
6. 模板	3			3	2		
7. 异常和智能指针	3				2		
8. C++标准模板库简介	5				3		
合 计	36			12	20		
总 计	48 学时+20 自学学时						

七、 课程目标达成途径及学生成绩评定方法

1. 课程目标达成途径

各个课程目标的达成途径如表 8 所示，可不仅限于此。

表 8 课程目标与达成途径

课程目标	达成途径
课程目标(1)：能够运用 C++语言的各种机制，具备阅读和调试 C++程序的基本能力。	以引导式、启发式和总结式教学方法为主，通过重点/难点内容讲解、课后作业、进行随堂提问、课堂程序演示等模式，帮助学生能够运用 C++语言的各种机制，具备阅读、理解和调试各类 C++程序的能力。
课程目标(2)：具备运用面向对象思维方式，为实际应用问题建立问题模型，设计合适 C++类的初步能力。	以启发式、分析式和研讨式教学方法为主，结合典型应用案例，通过上机实践、作业、课程设计、随堂提问、课堂讨论等模式，帮助学生具备运用面向对象思维方式，为实际应用

	问题建立问题模型，设计合适 C++类的初步能力。
课程目标(3)：能够综合运用 C++语言机制，设计结构良好、高效的现代 C++程序。	以引导式、启发式和总结式教学方法为主，通过重点/难点内容讲解、上机实践、课后作业、进行随堂提问、课堂讨论等模式，帮助学生掌握设计结构良好、高效的现代 C++程序，解决程序设计领域的复杂工程问题。
课程目标(4)：能够掌握面向对象程序设计的新技术与发展趋势，及时掌握国家相关方面的科技战略需求，树立强烈的爱国主义使命感与责任心。	以案例教学法、讲授法介绍课程的前沿技术，使学生进一步了解目前国内相关先进技术与取得的成就，从而建立强烈的民族自豪感与爱国主义使命感。

2. 学生成绩评定方法

该课程为考试课程，考试方式为闭卷。该课程采用形成性评价与终结性评价相结合的评价方法，学期总评成绩由二部分构成：平时成绩，占比 40%，任课教师可根据实际授课情况适当调整平时成绩组成比例；期末考试成绩，占比 60%。各部分的具体评价环节、关联课程目标、评价依据及方法和在总成绩中的占比，如表 9 所示。

表 9 课程考核与成绩评定方法

成绩构成	考核项目	考核关联的课程目标	考核依据与方法	占总评成绩的比重
平时成绩	课程思政实践	(4)	基于综合设计练习、前言介绍等主题，通过课堂展示、课堂小组讨论、网络学习等多种形式，考查学生对相关先进技术的了解情况以及核心价值观状况	5%
	课堂表现	(1)(2)(3)(4)	通过课堂讨论、问答、考勤等情况评定，占比不超过 10%	至少包含 2 项，共计占比 35%
	平时作业	(1)(2)(3)(4)	通过线上、线下作业完成情况考核，占比不超过 20%	
	课内上机	(1)(2)(3)(4)	课内上机作业是否及时完成、作业完成质量、验收情况，占比不超过 20%	
期末考试	闭卷考试	(1)(2)(3)(4)	考试成绩	60%
总评成绩		(1)(2)(3)(4)	=平时成绩*40%+考试成绩*60%	100%

八、教学资源

表 10 课程的基本教学资源

资源类型	资源
教材	李卫明, C++面向对象程序设计, 第 1 版, 西安电子科技大学出版社, 2020.5
参考书籍或文献	1.Stanley B. Lippman 著, 李师贤译, C++ Primer 中文版, 第 5 版, 电子工业出版社, 2013.9 2.Siddhartha Rao 著, 袁国忠译, 21 天学通 C++, 第 7 版, 人民邮电出版社, 2012.12 3.Bjarne Stroustrup 著, 裘宗燕译, C++程序设计语言, 特别版, 机械工业出版社, 2009.12 4.Nicolai M. Josuttis 著, 侯捷译, C++标准库, 第 2 版, 电子工业出版社, 2015.6

	5.郑莉, 董渊, 何江舟, C++语言程序设计, 第4版, 清华大学出版社, 2010.7
教学文档	无

九、 课程目标达成度定量评价

在课程结束后, 需要对每一个课程目标(含思政课程目标)进行达成度的定量评价, 用以实现课程的持续改进。

课程目标达成度的定量评价算法:

- 1、使用教学活动(如课程思政实践、课后作业、课堂练习、单元测验、演讲、课堂讨论、互动、验收成绩等等)成绩或期末考试部分题目得分率作为评价项目, 对某个课程目标进行达成度的定量评价;
- 2、为保证考核的全面性和可靠性, 要求对每一个课程目标的评价项目选择至少两种;
- 3、根据施教情况, 评价项目可以由教师自行扩展, 权重比例可以由教师自行设计;
- 4、对某一个课程目标有支撑的各评价项目权重之和为1;
- 5、使用所有学生(含不及格)的平均成绩计算。

本课程的课程目标达成度的定量评价算法建议如表 11 所示, 教师可根据授课方式及考核内容适当调整:

表 11. 课程目标达成度定量评价方法

课程目标	课程目标达成度评价方式
课程目标 1: 能够运用 C++语言的各种机制, 具备阅读和调试 C++程序的基本能力。	平时作业、课堂表现 0.2 课内上机 0.4 期末考试: 0.4
课程目标 2: 具备运用面向对象思维方式, 为实际问题建立问题模型, 设计合适 C++类的初步能力。	平时作业、课堂表现 0.3 课内上机 0.5 期末考试: 0.2
课程目标 3: 能够综合运用 C++语言机制, 设计结构良好、高效的现代 C++程序。	平时作业、课堂表现 0.2 课内上机 0.6 期末考试: 0.2
课程目标 4: 能够掌握面向对象程序设计的新技术与发展趋势, 及时掌握国家相关方面的科技战略需求, 树立强烈的爱国主义使命感与责任心。	课程思政 0.4 平时作业、课堂表现 0.2 课内上机 0.2 期末考试: 0.2

十、 说明

本课程大纲主要用于规范计算机类相关专业的《面向对象程序设计(C++)》课程的教学要求和教学规范，承担《面向对象程序设计(C++)》课程的教师须遵照本大纲安排授课计划、实施教学过程，完成学生学习成果评价、课程目标达成度评价。

本课程大纲自 2022 年开始执行，生效之日原先版本均不再使用。

十一、 编制与审核

表 14.大纲编制与审核信息

工作内容	责任部门或机构	负责人	完成时间
编制	程序设计课程组	李卫明	2022.02.25
审核	程序设计课程组	李强	2022.03.05
审定	计算机学院教学工作委员会	傅婷婷	2022.05.15