**《C++程序设计》课程教学大纲**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 英文课程名 | C++ Programming | | 总 学 时 | | 64 | 学 分 | 4 |
| 课程编码 | G126003 | | 理论教学学时 | | 48 | 适用专业 | 软件工程 |
| 课程类别（请在课程所属类别栏注明选修或必修） | 通识课程 |  | 实践  教学  学时 | 实验学时 |  | 先修课程 | 先修《程序设计基础C》  并修《离散数学》 |
| 大类基础课程 |  | 上机学时 | 16 | 开课学院（部） | 计算机科学与技术学院 |
| 专业课程 | 专业基础  必修课 | 其它 |  | 基层教学组织 | 计算机基础课程群教学团队 |

**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、教学目标**

**2.1 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。
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。
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。

**2.2 课程目标与毕业要求（指标点）对应关系**

该课程支撑以下毕业要求和具体细分指标点：

【毕业要求1】能够将数学、自然科学、工程基础和专业知识用于解决复杂软件工程问题。

支撑指标点1.3 能够将工程基础和专业知识用于求解软件领域复杂工程问题

【毕业要求3】能够设计针对复杂工程问题的解决方案，设计满足特定需求的系统、单元（部件）或工艺流程，并能够在设计环节中体现创新意识，考虑社会、健康、安全、法律、文化以及环境等因素

支撑指标点3.1：掌握解决复杂工程问题的程序设计语言基础。

【毕业要求5】使用现代工具：能够针对复杂工程问题，开发、选择与使用恰当的技术、资源、现代工程工具和信息技术工具，包括对复杂工程问题的预测与模拟，并能够理解其局限性

支撑指标点5.1：了解现代工程工具和信息技术工具的使用方法,了解实际计算机工程系统中各种现代工具的使用现状。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程目标 | 毕业要求指标点 | 教学环节 | | | | |
| 课堂授课 | 实验 | 作业 | 其他（如课堂讨论等） |
| **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。 | **指标点1.3** 能够将工程基础和专业知识用于求解软件领域复杂工程问题。  **指标点3.1**掌握解决复杂工程问题的程序设计语言基础。  **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,了解实际计算机工程系统中各种现代工具的使用现状。 | √ | √ | √ | √ |
| **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。 | **指标点3.1**掌握解决复杂工程问题的程序设计语言基础。  **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,了解实际计算机工程系统中各种现代工具的使用现状。 | √ | √ | √ | √ |
| **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。 | **指标点5.1**：了解实际计算机应用系统中现代工具、技术及资源的现状和使用方法 |  | √ | √ |  |

**三、课程教学内容及学时分配**

**1．理论教学安排**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 章节或知识点(模块) | 教学内容 | 学时分配 | 教学要求  (应明确教学重点、难点和教学方法) | 学生任务 | |
| 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。 | 2 | **教学重点：**  (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。  **教学难点：**  程序辅助描述工具，如流程图等的规范化。  **教学方法：**讲授+演示+例题讲解   * 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。 * 在开发环境中演示结构相关的例程 * 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。 | 可按书章节布置课后习题或者教师自行围绕该内容布置习题。 |  |
| 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。 | 4 | **教学重点：**  (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。  **教学难点：**理解和掌握链表的各种操作（建立、访问、删除、查找、逆转、合并等）。  **教学方法：**讲授+演示+例题讲解   * 讲授链表的基本定义方法和基本操作应用； * 在开发环境中演示链表的基本操作应用； | 可按书章节布置课后习题或者教师自行围绕该内容布置习题。 | 讨论：   * 链表操作的关键核心以及链表的变形？ |
| 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。 | 4 | **教学重点：**  (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。  **教学难点：**class的定义和使用，理解封装；与过程化程序设计方法对比。  **教学方法：**讲授+演示+例题讲解   * 讲授class的基本语法；类内成员函数在类内实现和类外实现的方法； * 讲授和演示基于类的面向对象程序的组织结构：自定义类库的组织方法和头文件引入 * 结合具体类的实例讲授public和private两种访问控制的差异。 | 可按书章节布置课后习题或者教师自行围绕该内容布置习题。 | 讨论：   * 类的语法以及类内的访问控制方式 |
| 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 。*** | 14 | **教学重点：**  (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。  **教学难点：**理解和掌握类的可缺省成员，明白可缺省成员的适用性；理解和掌握运算符的重载；理解和掌握构造析构的顺序以及数据成员的初始化顺序。  **教学方法：**讲授+演示+例题讲解   * 讲授类的４类可缺省成员函数以及他们的局限性，通过实例演示在什么情况下缺省的够用，而在另一些情况下需要自己重写可缺省的部分； * 演示和讲解const和static在类内的使用情况，对成员函数和数据成员的约束； * 讲解friend的使用； * 通过实例讲解演示类的构造和析构顺序，数据成员的初始化顺序； * 讲解对象指针，类的引用，赋值及转型的概念。 | 可按书章节布置课后习题或者教师自行围绕该内容布置习题。 | 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？ |
| 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。 | 6 | **教学重点：**  (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；  (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；  **教学难点：**理解和掌握继承模型；继承模型下的构造和析构次序；  **教学方法：**讲授+演示+例题讲解   * 讲授继承现象，Ｃ＋＋中的继承模型； * 结合实例讲授public,   protected,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；   * 演示继承结构中构造和析构的次序； * 介绍定义新类的更可靠的方法：继承和组合； | 可按书章节布置课后习题或者教师自行围绕该内容布置习题。 | 讨论：   * 继承和组合有什么不一样？ |
| 6 | 类的多态性 | 类的多态性概念  类的抽象  虚函数  ***多态编程*** | 6 | **教学重点：**  (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；进一步掌握动态多态；  **教学难点：**虚函数的语法及使用。  **教学方法：**讲授+演示+例题讲解   * 结合实例演示并讲解不同事物同名操作下的不同实现。 * 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。 * 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。 | 可按书章节布置课后习题或者教师自行围绕该内容布置习题。 | 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？ |
| 7 | 抽象  方法 | 抽象基类  虚基类与纯虚函数 | 6 | **教学重点：**  (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。  **教学难点：**理解和掌握虚基类和纯需基类的定义方法和差异；理解动态多态的概念。  **教学方法：**讲授+演示+例题讲解   * 结合实例演示并讲解虚基类的定义及使用方法； * 进一步讲授纯虚函数的定义语法和含义，以及应用茶馆和； * 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； * 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； | 可按书章节布置课后习题或者教师自行围绕该内容布置习题。 | 讨论：   * 抽象基类是否可以实例化？ * 纯虚函数有什么特点？ |
| 8 | ***模板与异常处理*** | ***函数模板与模板类***  ***异常处理*** | 6 | **教学重点：**  (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。  **教学难点:**理解和掌握模版的概念，会定义和使用类模版；理解和掌握基本异常处理处理模型。  **教学方法：**讲授+演示+例题讲解   * 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； * 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。 | 可按书章节布置课后习题或者教师自行围绕该内容布置习题。 | 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？ |

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 | 项目名称 | 学时 | 类型 | 每组人数 | 教学要求  (应明确教学重点、难点和教学方法) | 学生任务 | |
| 作业要求 | 其他要求(自学/讨论） |
| 1 | 单链表创建及其他操作 | 2 | 设计 | 1 | **教学要求：**   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   **教学重点：**  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；  **教学难点：**  综合使用链表，并能进行程序的调试测试和结果分析  **教学方法：**讲授+任务驱动式教学+上机实践 | 上交实验成果及相关文档 |  |
| 2 | 类与对象1：类的声明、定义和使用 | 2 | 验证  设计 | 1 | **教学要求：**   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   **教学重点：**  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；  **教学难点：**  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；  **教学方法：**讲授+任务驱动式教学+上机实践 | 上交实验成果及相关文档 |  |
| 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 | 验证  设计 | 1 | **教学要求：**   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   **教学重点：**  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；  **教学难点：**  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；能正确使用赋值运算重载（类内成员）的形式；  **教学方法：**讲授+任务驱动式教学+上机实践 | 上交实验成果及相关文档 |  |
| 4 | 类与对象3：类的操作符重载 |  | 设计 | 1 | **教学要求：**   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   **教学重点：**   1. 掌握类的操作符重载，能够编程实现操作符重载； 2. 能够掌握在类内重载和在类外作为普通函数重载的语法；   **教学难点：**  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；能够掌握在类内重载和在类外作为普通函数重载的语法；  **教学方法：**讲授+任务驱动式教学+上机实践 | 上交实验成果及相关文档 |  |
| 5 | 类的继承1：类的继承及派生类使用 | 2 | 设计 | 1 | **教学要求：**   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   **教学重点：**  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。  **教学难点：**  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。  **教学方法：**讲授+任务驱动式教学+上机实践 | 上交实验成果及相关文档 |  |
| 6 | 类的多态性1：虚函数的使用 | 2 | 设计 | 1 | **教学要求：**   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   **教学重点：**  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  **教学难点：**  能够根据问题模型自行设计简单的继承层次用于求解；  **教学方法：**讲授+任务驱动式教学+上机实践 | 上交实验成果及相关文档 |  |
| 7 | 类的多态性 | 2 | 设计 | 1 | **教学要求：**  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  **教学重点：**  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；  **教学难点：**  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；  **教学方法：**讲授+任务驱动式教学+上机实践 | 上交实验成果及相关文档 |  |
| 8 | **一个复杂的类定义及实现（*类模版）*** | 2 | 设计 | 1 | **教学要求：**   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   **教学重点：**  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；  **教学难点：**  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；可以完善程序，使用基本的异常处理；  **教学方法：**讲授+任务驱动式教学+上机实践 | 上交实验成果及相关文档 |  |

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。

**四、考核方式及成绩评定方式**

本课程的考核强调课堂教学成果与学生上机实践的综合考核。学生成绩由考试成绩(50%)和过程化考核成绩(50%)构成；其中过程化考核成绩内容要求实验和作业不低于40%，其中实验不低于20%，允许有其他形式参与过程化考核，如课堂提问，小组讨论等，该项比例不高于10%。

作业方面要重点考核学生掌握课堂学习的基础知识以及知识点运用的程度，课内上机实验考察学生掌握运用所学知识点进行编程实践的能力。课堂表现主要从学生上课是否专心听讲、回答教师提问是否正确，以及分组讨论是否积极、正确、有独特见解等，以提高课堂教学效果和运用工程基础知识和本专业基本理论解决实际工程问题的能力。

期末考试将按照本课程的教学目标全面考核学生的课程学习效果，分析课程对毕业要求的达成度。

**五、教材、课程网址及参考书目**

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

课程在学校网络教学平台的地址(核心课程必填)： http://i.mooc.chaoxing.com

参考书：

1. 《C++程序设计教程（第二版）》，钱能，清华大学出版社，2005
2. 《C++程序设计教程（第二版）习题与解答》，钱能著，清华大学出版社，2009.7
3. 《C++Primer Plus（第五版）中文版》，[[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，人民邮电出版社，2005.1
4. 《C++标准程序库》，侯杰，孟岩译，华中科技大学出版社，2002.9
5. C++ Primer（第五版）. Stanley B.Lippman，Josee Lajoie．，王刚等译．电子工业出版社，2013.
6. 《C++编程思想（第一卷）》，Bruce Eckel著，机械工业出版社，2002.9

**执笔者：毛国红、王英姿**

**审核者：田贤忠、陈志扬**

**课程教学团队成员：陈志扬、江颉、赖梅、刘文捷、毛国红、钱能、田贤忠、王英姿**