**《数据结构》课程教学大纲**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 英文课程名 | Data structures | | 总 学 时 | | 64 | 学 分 | 4 |
| 课程编码 | G126085 | | 理论教学学时 | | 48 | 适用专业 | 软件工程专业 |
| 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 | C++程序设计、离散数学 |
| 大类基础课程 | 必须 | 上机学时 | 16 | 开课学院（部） | 计算机学院 |
| 专业课程 |  | 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

数据结构是计算机学科的基础理论课，是计算机学科的核心课程之一，是计算机系统应用开发的重要基础理论。本课程介绍各种线性和非线性数据结构在计算机中的存储组织和访问操作方法，并对查找、排序、递归等相关算法进行详细介绍。主要数据结构包括：向量、链表、列表、栈、队列、二叉树、堆、图等。课程采用面向对象的方法来描述数据结构，并用C++语言作为算法的描述语言和开发工具，强化数据结构基本知识和面向对象程序设计能力的训练，掌握好该课程将为后续计算机专业课程打下坚实的基础。

二、教学目标

2.1 课程教学目标

本课程的教学目标是使学生掌握数据结构和算法的基本理论知识，培养学生掌握面向对象的设计理论与方法，具备C++的软件开发能力，能够运用工程基础知识和本专业基本理论(包括常用的数据结构和算法)解决复杂计算机工程问题。具体如下：

1. 了解数据结构的基本概念，掌握主要数据结构的存储组织和访问操作方法，包括向量、链表、列表、栈和队列、二叉树、堆、图等，从而掌握解决复杂计算机工程问题所需的数据结构理论基础。
2. 掌握主要的查找算法、排序算法，以及递归等的使用，了解算法的时间复杂度分析，从而掌握解决复杂计算机工程问题所需的算法理论基础。
3. 培养学生进行复杂程序的设计能力和编程能力，能采用C++语言来解决复杂的计算机工程问题，掌握面向对象的程序设计方法，养成良好的编程习惯，从而掌握解决复杂计算机工程问题所需的程序设计能力和编程能力。
4. 培养学生使用数据结构和算法来解决复杂计算机工程问题，从而掌握解决复杂计算机工程问题所需的实践能力。

2.2课程目标与毕业要求（指标点）的对应关系

该课程支撑以下毕业要求和具体细分指标点：

**【毕业要求1】**工程知识：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。

**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1.4**：能够将数学、自然科学、工程基础和专业知识用于对求解结果进行分析和评价。

**【毕业要求2】**问题分析：能够应用数学、自然科学和计算机工程科学的基本原理，识别、表达、并通过文献研究分析复杂计算机工程问题，以获得有效结论。

**指标点2.4**：能够对多种解决途径的可行性和有效性进行对比，以得出有效方法和结论。

**【毕业要求4】**研究：能够基于科学原理并采用科学方法对复杂软件工程问题进行研究，包括设计实验、分析与解释数据、并通过信息综合得到合理有效的结论。

**指标点4.1**：具有基于科学原理和科学方法设计软件实验项目的能力。

本课程目标与毕业要求（指标点）的对应关系如表1所示。

表1 课程目标与毕业要求（指标点）的对应关系

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程目标 | 毕业要求指标点 | 教学环节 | | | | |
| 课堂授课 | 上机 | 作业 | 课堂讨论 |
| 目标1：了解数据结构的基本概念，掌握主要数据结构的存储组织和访问操作方法，包括向量、链表、列表、栈和队列、二叉树、堆、图等，从而掌握解决复杂计算机工程问题所需的数据结构理论基础。 | 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1.4：能够将数学、自然科学、工程基础和专业知识用于对求解结果进行分析和评价。  指标点2.4：能够对问题的各种解决途径的可行性和有效性进行对比，以得出有效结论。 | √ | √ | √ | √ |
| 目标2：掌握主要的查找算法、排序算法，以及递归等的使用，了解算法的时间复杂度分析，从而掌握解决复杂计算机工程问题所需的算法理论基础。 | 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1.4：能够将数学、自然科学、工程基础和专业知识用于对求解结果进行分析和评价。  指标点2.4：能够对问题的各种解决途径的可行性和有效性进行对比，以得出有效结论。 | √ | √ |  | √ |
| 目标3：掌握面向对象的程序设计方法（即类的设计思路），培养学生进行复杂程序的设计能力、编程能力、以及调试和分析能力，能采用C++语言编程实现特定要求的程序，从而掌握解决复杂计算机工程问题所需的程序设计能力和编程能力。 | 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。 | √ | √ | √ |  |
| 目标4：培养学生应用一种或多种数据结构和算法来解决复杂工程问题的能力，培养学生选择合适的数据结构和算法来解决复杂计算机工程问题的能力，从而掌握解决复杂计算机工程问题所需的工程实践能力。 | 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1.4：能够将数学、自然科学、工程基础和专业知识用于对求解结果进行分析和评价。  指标点2.4：能够对问题的各种解决途径的可行性和有效性进行对比，以得出有效结论。  指标点4.1：具有基于科学原理和科学方法设计软件实验项目的能力。 | √ | √ |  | √ |

三、课程教学内容及学时分配

3.1 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 | 第一章  绪论 | 1. 数据类型和数据结构 2. 数据结构结构的基本概念：逻辑结构和物理存储结构 3. 数据结构的抽象数据类型表示 4. 数据结构的应用 | 2 | 教学要求：   1. 学生了解数据结构的概念、抽象数据类型表示。 2. 培养学生掌握基本知识的能力。   教学重点：数据结构概念。  教学难点：数据结构概念。  教学方法：讲授 + 讨论 | 1. 完后课后编程作业第2题。   （作业内容可调整） | 自学：要求学生课后自学动态数据空间分配方法等内容。  讨论：针对如何设计类、抽象数据这一专题，安排学生3人一组进行讨论。 |
| 2 | 第二章  动态数组（向量） | 1. 数组的存储与访问方式 2. 动态数组的空间申请和空间管理 3. 动态数组的声明和实现 4. 动态数组的模板实现方法\* 5. Vector的应用 6. String的基本方法和应用 7. 模式匹配方法\* | 4 | 教学要求：   1. 学生掌握动态数组的基本构建方法； 2. 理解动态数组的空间申请与管理方法； 3. 掌握vector的应用方法； 4. 理解模板的概念和使用方法； 5. 培养学生掌握基本知识的能力，以及程序设计能力和编程能力。   教学重点：   1. 动态数组。   教学难点：动态数组  教学方法：讲授 + 讨论+ 任务驱动式教学。 | 1. 完后课后编程练习； 2. 增加向量的插入和删除方法； 3. 增加向量构造函数； 4. 实现向量重复元素的删除。   （作业内容可调整） | 自学：要求学生课后自学模式匹配算法。  讨论：针对深拷贝、浅拷贝专题，安排学生3人一组进行讨论。 |
| 3 | 第三章  链表 | 1. 链表的基本结构和创建方式 2. 链表的遍历 3. 链表的插入和删除 4. 链表的归并和拆分 5. 链表应用 6. 列表的三种实现方式（数组、向量、链表） | 6 | 教学要求：   1. 掌握链表的基本构建、遍历、插入和删除方法； 2. 掌握链表的合并与拆分方法，能够应用链表解决简单工程问题。 3. 培养学生掌握基本知识的能力，以及程序设计能力和编程能力。   教学重点：链表  教学难点：链表  教学方法：讲授 + 讨论+ 任务驱动式教学 | 1. 完后课后编程练习：4,7,12； 2. 实现链表中重复元素的删除。   （作业内容可调整） | 自学：要求学生课后自学链表池设计方法。  讨论：针对如何反转链表这一专题，安排学生3人一组进行讨论。 |
| 4 | 第四章  栈 | 1. 栈的基本结构和创建方式（数组、向量、链表） 2. 栈的基本属性和基本操作 3. 栈的应用 4. 中缀转后缀算法/后缀表达式计算 | 4 | 教学要求：   1. 掌握栈的基本创建、压栈、出栈方法； 2. 掌握栈的不同存储结构实现方法； 3. 理解栈的基本属性； 4. 掌握利用栈特性实现中缀转后缀算法； 5. 培养学生掌握基本知识的能力，程序设计能力和编程能力，以及解决复杂工程问题的实践能力。   教学重点：栈  教学难点：栈  教学方法：讲授 + 讨论+自主学习 | 1. 完后课后习题： 20,21,26, 27,31, 34。   （作业内容可调整） | 自学：要求学生课后自学系统栈的应用。  讨论：针对如何实现浮点数的中缀转后缀算法这一专题，安排学生3人一组进行讨论。 |
| 5 | 第五章  队列 | 1. 队列的基本结构和创建方式（数组、向量、链表） 2. 队列的基本属性和基本操作 3. 队列的应用 4. 优先队列 | 4 | 教学要求：   1. 掌握队列的基本创建、入队、出队方法； 2. 掌握队列的不同存储结构实现方法； 3. 理解队列的基本属性； 4. 掌握优先队列实现方法； 5. 培养学生掌握基本知识的能力，程序设计能力和编程能力，以及解决复杂工程问题的实践能力。   教学重点：队列  教学难点：队列  教学方法：讲授 + 讨论+自主学习 | 1. 完后课后编程练习；实现队列中指定元素删除。   （作业内容可调整） | 自学：要求学生课后自学STL中deque的操作方法。  讨论：针对如何实现离散事件动态模拟这一专题，安排学生3人一组进行讨论。 |
| 6 | 第六章  双向链表 | 1. 双向链表的基本结构和创建方式 2. 双向链表的遍历、插入和删除方法 3. 双向链表的应用 4. list容器 5. list迭代器的使用方法和构造原理\* | 4 | 教学要求：   1. 掌握双向链表的基本创建、遍历、插入和删除方法； 2. 理解循环链表的结构和访问方法； 3. 理解迭代器的原理和使用方法； 4. 培养学生掌握基本知识的能力，程序设计能力和编程能力，以及解决复杂工程问题的实践能力。   教学重点：双向链表  教学难点：双向链表  教学方法：讲授+ 讨论+任务驱动式教学 | 1. 完后课后编程练习：10,11,实现双向链表中重复元素的删除。   （作业内容可调整） | 自学：要求学生课后自学STL中list的操作方法。  讨论：针对如何实现迭代器这一专题，安排学生3人一组进行讨论。 |
| 7 | 第七章  递归算法 | 1. 递归算法的基本概念和实现方法 2. 数制转换 3. 汉诺塔 | 2 | 教学要求：   1. 掌握递归算法的基本实现方法； 2. 理解递归算法的调用方式和终止方式； 3. 理解递归算法的解题思路。 4. 培养学生掌握基本知识的能力，程序设计能力和编程能力。   教学重点：递归  教学难点：递归  教学方法：讲授 + 直观演示+ 自主学习 | 1. 完后课后编程练习：单链表反转的递归实现。   （作业内容可调整） | 自学：要求学生课前自学汉诺塔的基本思路。  讨论：针对递归深度这一专题，安排学生3人一组进行讨论。 |
| 8 | 第八章  二叉树 | 1. 二叉树的基本概念和表示方法 2. 二叉树的创建、递归遍历方法 3. 二叉树的非递归遍历方法 4. 二叉搜索树的创建、插入、删除方法 5. 二叉平衡树 | 8 | 教学要求：   1. 掌握二叉树的基本创建、递归遍历、非递归遍历方法； 2. 掌握二叉搜索树的插入和删除方法； 3. 掌握二叉平衡树的调整方法。 4. 培养学生掌握基本知识的能力，程序设计能力和编程能力，以及解决复杂工程问题的实践能力。   教学重点：二叉树  教学难点：二叉树  教学方法：讲授 +讨论+自主学习+任务驱动式教学 | 1. 完后课后练习题：4,8,12,13, 14,15,16,20,21   （作业内容可调整） | 自学：要求学生课后自学森林及森林与二叉树的转换方法。  讨论：针对二叉树旋转90度递归打印、正向递归打印这一专题，安排学生3人一组进行讨论。 |
| 9 | 第九章  堆 | 1. 完全二叉树结构 2. 堆的创建方法、堆调整 3. 堆的插入、删除方法 4. 堆的应用－优先队列 | 2 | 教学要求：   1. 掌握堆的基本创建、插入、删除等方法，理解堆调整算法； 2. 掌握最大堆、最小堆创建方法； 3. 培养学生掌握基本知识的能力，程序设计能力和编程能力，以及解决复杂工程问题的实践能力。   教学重点：堆  教学难点：堆  教学方法：讲授 +讨论+任务驱动式教学 | 1. 完后课后练习题：1,2,3,6,7, 12, 15   （作业内容可调整） | 自学：要求学生课后自学堆排序方法。  讨论：针对堆的链表实现这一专题，安排学生3人一组进行讨论。 |
| 10 | 第十章  霍夫曼树 | 1. 编码压缩方法 2. 霍夫曼编码 3. 霍夫曼树创建方法 | 2 | 教学要求：   1. 掌握霍夫曼树的基本创建方法和霍夫曼编码方法。 2. 培养学生掌握基本知识的能力。   教学重点：霍夫曼编码  教学难点：霍夫曼编码  教学方法：讲授 + 自主学习 | 1. 完后课后练习题：1,2,3,4,6   （作业内容可调整） | 自学：要求学生课后自学霍夫曼树创建算法的编程实现。  讨论：针对最优二叉树这一专题，安排学生3人一组进行讨论。 |
| 11 | 第十一章  复杂树结构\* | 1. 2-3-4树创建方法\* 2. B树创建、插入和删除方法\* | 2 | 教学要求：   1. 掌握、理解B树的创建、分裂和合并算法。 2. 培养学生掌握基本知识的能力。   教学重点：B树  教学难点：B树  教学方法：讲授 + 讨论 | 1. 完后课后练习题：2,4,5,8   （作业内容可调整） | 自学：要求学生课后自学红黑数的创建、更新方法。  讨论：针对B树、B+树这一专题，安排学生3人一组进行讨论。 |
| 12 | 第十二章  查找算法 | 1. 顺序查找算法及时间复杂度 2. 折半查找算法及时间复杂度 3. 哈希表构造方法（开散列、闭散列） 4. 哈希查找算法及冲突解决方法 | 2 | 教学要求：   1. 掌握各种基本查找算法及时间复杂度； 2. 掌握哈希表的两张构造方法及相应的查找算法； 3. 理解查找长度的计算方法。 4. 培养学生掌握基本知识的能力，程序设计能力和编程能力，以及解决复杂工程问题的实践能力。   教学重点：查找算法  教学难点：查找算法  教学方法：讲授 + 讨论 +任务驱动式教学 | 1. 完后课后练习题：7,9   （作业内容可调整） | 自学：要求学生课后自学哈希函数构造方法。  讨论：针对平均成功查找长度这一专题，安排学生3人一组进行讨论。 |
| 13 | 第十三章  排序算法 | 1. 简单排序算法及时间复杂度（选择排序、交换排序、冒泡排序、插入排序） 2. 高级排序算法及时间复杂度（快速排序、堆排序、归并排序） | 4 | 教学要求：   1. 掌握各种基本排序算法及时间复杂度； 2. 理解算法的最好排序时间、最坏排序时间及平均排序时间。 3. 培养学生掌握基本知识的能力，程序设计能力和编程能力，以及解决复杂工程问题的实践能力   教学重点：排序算法  教学难点：排序算法  教学方法：讲授 + 讨论 +任务驱动式教学 | 1. 完后课后练习题：2,5,6,7,9   （作业内容可调整） | 自学：要求学生课后自学希尔排序算法。  讨论：针对快速排序算法改进这一专题，安排学生3人一组进行讨论。 |
| 14 | 第十四章  图结构 | 1. 图的基本结构（无向图和有向图） 2. 图的基本表示方法 3. 图的基本遍历算法（深度、广大遍历）、插入和删除算法 4. 最短路径算法 5. 最小生成树算法 | 2 | 教学要求：   1. 掌握图的基本表示方法和图的两种遍历方法； 2. 掌握图的应用算法－最短路径算法、最小生成树算法等。 3. 培养学生掌握基本知识的能力。   教学重点：最短路径算法  教学难点：最短路径算法  教学方法：讲授 + 讨论 | 1. 完后课后练习题：3,5,9,21, 22,38   （作业内容可调整） | 自学：要求学生课后自学Floyd等算法。  讨论：针对图最短路径算法这一专题，安排学生3人一组进行讨论。 |

注：\*表示可选讲内容。

3.2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 | 数组/Vector对比 | 2 | 验证型 | 1 | 教学要求：   1. 要求学生熟悉C++编程语言和C++开发环境(如VC++6.0或VS2012)； 2. 熟悉STL中的vector容器； 3. 根据要求编程实现两个函数，分别采用数组和vector作为参数，进行对比分析；   教学重点：数组和vector的不同点  教学难点：数组和vector的不同点  教学方法：任务驱动式教学+操作实践+课堂讨论 | 1. 完成代码设计和实现 2. 完成实验报告 | 自学：   1. 熟悉软件开发和调试平台，自学相关资料； 2. 根据实验要求，进行需求分析，程序设计、代码实现和调试测试等。   讨论：分析数组和vector的主要异同点，及应用场合。 |
| 2 | 无序表生成有序表，以及两个有序表合并成一个新的有序表 | 4 | 设计型 | 1 | 教学要求：   1. 要求学生根据需求实现一个链表类； 2. 实现链表排序、有序链表合并等多种方法。 3. 要求采用面向对象的设计方法，设计的类可以满足复杂工程问题的需求。   教学重点：链表类的设计  教学难点：链表类的设计  教学方法：任务驱动式教学+操作实践+课堂讨论 | 1. 完成代码设计和实现 2. 完成实验报告 | 自学：   1. 熟悉软件开发和调试平台，自学链表相关资料； 2. 根据实验要求，进行需求分析，程序设计、代码实现和调试测试等。   讨论：链表类的主要成员函数及其实现。 |
| 3 | 设计实现双端栈 | 2 | 设计型 | 1 | 教学要求：   1. 要求学生选择合适的数据结构，设计一个类，实现双端栈的主要功能； 2. 要求采用面向对象的设计方法，设计的类可以满足复杂工程问题的需求。   教学重点：双端栈类的设计  教学难点：双端栈类的设计  教学方法：任务驱动式教学+操作实践+课堂讨论 | 1. 完成代码设计和实现 2. 完成实验报告 | 自学：   1. 熟悉软件开发和调试平台，自学双端栈相关资料； 2. 根据实验要求，进行需求分析，程序设计、代码实现和调试测试等。   讨论：双端栈的主要成员函数及其实现。 |
| 4 | 设计实现优先队列 | 2 | 设计型 | 1 | 教学要求：   1. 要求学生选择合适的数据结构，设计一个类，实现优先级队列； 2. 要求采用面向对象的设计方法，设计的类可以满足复杂工程的需求。   教学重点：优先级队列类的设计  教学难点：优先级队列类的设计  教学方法：任务驱动式教学+操作实践+课堂讨论 | 1. 完成代码设计和实现 2. 完成实验报告 | 自学：   1. 自学优先级队列相关资料； 2. 根据实验要求，进行需求分析，程序设计、代码实现和调试测试等。   讨论：优先级队列的主要成员函数及其实现。 |
| 5 | 多项式合并 | 2 | 设计型 | 1 | 教学要求：   1. 要求学生选择合适的数据结构，设计一个类，实现多项式合并； 2. 要求采用面向对象的设计方法，设计的类可以满足复杂工程的需求。   教学重点：多项式类的设计  教学难点：多项式类的设计  教学方法：任务驱动式教学+操作实践+课堂讨论 | 1. 完成代码设计和实现 2. 完成实验报告 | 自学：   1. 自学优先级队列相关资料； 2. 根据实验要求，进行需求分析，程序设计、代码实现和调试测试等。   讨论：优先级队列的主要成员函数及其实现。 |
| 6 | 表达式二叉树 | 4 | 设计型 | 1 | 教学要求：   1. 要求学生选择合适的数据结构，设计一个类，实现二叉树的主要属性和方法； 2. 要求采用面向对象的设计方法，设计的类可以满足复杂工程的需求。   教学重点：二叉树类的设计  教学难点：二叉树类的设计  教学方法：任务驱动式教学+操作实践+课堂讨论 | 1. 完成代码设计和实现 2. 完成实验报告 | 自学：   1. 自学二叉树相关资料； 2. 根据实验要求，进行需求分析，程序设计、代码实现和调试测试等。   讨论：二叉树类的主要成员函数及其实现。 |
| 9 | 课程项目 | 课外完成 | 综合  设计型 | 1-3 | 教学要求：   1. 要求学生综合运用课程所学的数据结构和算法，实现和解决特定的复杂工程问题； 2. 完成撰写相关技术文档，学生对复杂软件工程问题进行面向对象设计，将复杂问题形式化描述，具有一定的软件开发能力。   教学重点：类的设计  教学难点：类的设计  教学方法：任务驱动式教学+操作实践+课堂讨论 | 1. 完成代码设计和实现 2. 完成实验报告 | 自学：   1. 自学相关资料； 2. 根据实验要求，进行需求分析，程序设计、代码实现和调试测试等。   讨论：如何运用软件工程思想来解决复杂工程问题。 |

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，但需保证不少于16个学时，且覆盖全部目标和毕业要求指标点。

四、考核方式及成绩评定方式

该课程的考核强调过程化考核。其总成绩分为进程性成绩（占50%）和期末成绩（占50%）两部分。进程性成绩主要考核学生的作业、课内实验、课堂表现、课堂小测等多个方面。各部分的基本要求如下：

（1）期末考试：占总成绩的50%。要求：试卷难度适中，选择、名词解释、简答等基础性题目严格控制比例，加大综合性题目的比例，重在考查学生运用知识解决复杂工程问题的能力。

（2）作业：占总成绩的20%。要求教师针对某些知识模块布置一定数量的课后作业或课外编程题，以巩固知识或拓展思维。对于作业中的共性问题，教师须在课堂讲解，以帮助学生提高和进步，考核学生掌握基础知识的能力和程序设计能力等。

（3）课内实验：占总成绩的20%。要求教师根据授课内容，布置一定的课内实验任务，要求学生熟悉相应的开发工具，查找相关资料，学习程序设计和编程及调试分析等；同时要求学生学会如何利用网络查找解决编程过程中出现的问题。教师可以在实验课内解答学生的问题，并根据学生的完成情况，给予适当指导和督促，考核学生的程序设计能力、编程能力以及综合实践能力。

（4）其它：占总成绩的10%。包括课堂表现、期中考试、课堂小测等。要求教师根据授课内容、学生讨论情况、回答问题情况、期中考试或课堂小测等考核学生对于知识的掌握和程序设计能力等

期末考试采用闭卷形式，主要考核学生掌握基础知识的程度、综合计算和运用能力、以及程序设计和编程能力。最终，将按照本课程的教学目标全面考核学生课程学习的效果，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：“数据结构与算法分析－C++语言描述（第2版）”，Larry Nyhoff著，黄达明等译，清华大学出版社出版，2006年11月，2016年10月第7次印刷

参考书：

1. 数据结构与算法分析（C++版）（第三版），Clifford A. Shaffer著，电子工业出版社出版，2013年10月.
2. Data Structures with C++ Using STL, William Ford著,清华大学出版社, 2003年1月第1版, 2003年5月；
3. 数据结构：C++语言描述，苏仕华,刘燕君,刘振安著，机械工业出版社，2014年01月
4. 数据结构(C++版)(第3版)，叶核亚著，电子工业出版社，2014年02月
5. 数据结构（C++语言版）（第3版），作者:邓俊辉，清华大学出版社，2013年09月

**执笔者：刘端阳**

**审核者：李曲**

**课程教学团队成员：刘志、潘翔、陈志杨、黄伟、王丽萍**

**毛国红、刘端阳、赖梅、张晖**