1. 教学目标

算法与数据结构课程设计配合“算法与数据结构”理论课程的学习，培养学生将现实问题转换为逻辑模型和程序等解决问题的能力，使学生能在熟练掌握算法与数据结构理论知识的基础上使用计算机解决各种应用问题，为后续专业课程的学习打下坚实的问题求解能力和动手实践能力。具体包括：

（1）熟练掌握线性表、字符串、栈和队列以及树的应用，了解并掌握数据结构与算法的设计方法，具备初步的独立分析和设计能力。（支撑毕业要求1、2、5）

（2）掌握散列、索引、简单排序、快速排序、堆排序及归并排序等算法思想及实现方法，初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能。（支撑毕业要求1、2、4、5）

（3）掌握图的广度、深度优先遍历、最小生成树及其构造、最短路径算法以及拓扑序、关键路径算法等提高综合运用所学的理论知识和方法独立分析和解决问题的能力。（支撑毕业要求1、2、4、5、10）

（4）培养学生的实践能力、创新能力和团队合作与交流能力。训练学生用系统的观点和软件开发一般规范进行软件开发，培养软件工作者所应具备的科学的工作方法和作风。（支撑毕业要求1、2、4、12）

1. 课程要求

本课程设计实行互动研究型教学，在问题分析、数据结构与算法的设计和编程实现等环节对学生不断进行强化训练，要求学生独立完成基本实验，采用小组合作分工模式完成课程设计综合选题。本课程的具体要求为：

（1）独立完成基本实验项目，及时与教师沟通程序设计中遇到的各种问题；

（2）每3~4人一个小组，根据课程设计要求，对选题进行需求说明分析、功能模块设计、软件调试和测试，编写课程设计报告。

1. 教学内容

1、问题分析和任务定义。根据设计题目的要求，充分地分析和理解问题，明确问题要求做什么？（而不是怎么做？）限制条件是什么？

2、逻辑设计。对问题描述中涉及的操作对象定义相应的数据类型，并按照以数据结构为中心的原则划分模块，定义主程序模块和各抽象数据类型。逻辑设计的结果应写出每个抽象数据类型的定义(包括数据结构的描述和每个基本操作的功能说明)，各个主要模块的算法，并画出模块之间的调用关系图。

3、物理设计。定义相应的存储结构并写出各函数的伪码算法。在这个过程中，要综合考虑系统功能，使得系统结构清晰、合理、简单和易于调试，抽象数据类型的实现尽可能做到数据封装，基本操作的规格说明尽可能明确具体。详细设计的结果是对数据结构和基本操作作出进一步的求精，写出数据存储结构的类型定义，写出函数形式的算法框架。

4、程序编码。把详细设计的结果进一步求精为程序设计语言程序。同时加入一些注解和断言，使程序中逻辑概念清楚。

5、程序调试与测试。采用自底向上，分模块进行，即先调试低层函数。能够熟练掌握调试工具的各种功能，设计测试数据确定疑点，通过修改程序来证实它或绕过它。调试正确后，认真整理源程序及其注释，形成格式和风格良好的源程序清单和结果。

6、结果分析。程序运行结果包括正确的输入及其输出结果和含有错误的输入及其输出结果。算法的时间、空间复杂性分析。

7、编写课程设计报告。

1. 考核方式

课程设计结束时，要求学生提交课程设计报告（附源程序），可运行的软件系统。从程序设计结果、作品设计方案、报告撰写和答辩表现等四方面进行考核评分，以优秀、良好、中、及格、不及格五等给出成绩。