



信息工程系

教

案

课程名称： 程序设计基础

教 师： 林晓红

总 学 时： 60

理论学时： 30

实训学时： 30

上课班级： 物联网应用技术 251/（自主）251

授课学期： 2025-2026 第一学期

揭阳职业技术学院 教案

课程名称	程序设计基础		
课程类型	专业课	开课单位	信息工程系
授课教师	林晓红	职称	讲师
授课专业	物联网应用技术	年级	2025
授课方式	理论+实践	考核方式	考试
课程教学总学时数	60 学时	学分数	3 学分
学时分配	课堂讲授：30 学时； 实验：30 学时		
教材名称	C 语言程序设计实例教程（慕课版）第 3 版）		
作者	常中华 王春雷 毛旭婷 陈静		
出版社	人民邮电出版社	出版时间	2023 年 11 月第 3 版
授课时间	2025~2026 学年度第一学期		
课程简介	本课程主要向学生讲授 C 语言程序设计基础知识		

单元名称	单元 1 初识 C 语言		
授课类型	<input type="checkbox"/> A 类纯理论课	<input checked="" type="checkbox"/> B 类理论+实践	<input type="checkbox"/> C 类纯实践课
学时	理论 2、实验 2		
课型	新授		
课程思政	在文字未产生时，中国古代就有“上古结绳而治”的说法（《周易·系辞》），这是一种古老的计数和记事方法，说明中国在人类文明发展史上的先进、智慧和创新。在阿拉伯数字出现前，中国古代就发明了“算盘”计算工具，还传播到了世界其他地区，对人类的计算方式和数学发展产生了深远的影响。		
教学重点及难点 学生学习条件分析	1、起点分析 （1）具备高中毕业生知识水平 （2）具备计算机基本操作能力、具备软件安装与卸载能力。 2、重点分析 （1）了解 C 语言的历史和发展 （2）掌握 C 语言的基本结构 （3）掌握 DEV-C++集成环境的应用 3、难点分析 （1）C 语言的基本结构 （2）C 语言编程步骤		
教学方法手段	1、教学方法：采用讲授法、课堂讨论法、练习法、实训法、演示法。 2、学习方法：合作学习，实训学习，自主学习。 3、教学手段：多媒体，项目演示。		
教学资源	1、学校资源：多媒体教室、实训教室。 2、文献资源：参考教材，课程大纲，课外读物，实训要求。 3、网络资源：学习通、二维码扫描视频。		
能力指标及课程教学目标	1、能够使用 DEV-C++集成环境编写一个 C 语言源程序，输出一行字符。 2、能够使用初步读懂并编写运行一个有简单输入、计算、输出功能的程序。		
单元教学目标	1、知识目标 （1）了解 C 语言的历史和发展 （2）掌握 C 语言的基本结构 （3）掌握 DEV-C++集成环境的应用 2、能力目标 （1）能够使用 DEV-C++集成环境编写一个 C 语言源程序。 （2）能够使用初步读懂并编写运行一个有简单输入、计算、输出功能的程序。 3、素养目标 （1）建立初步编程解决问题的思想。 （2）具有独立解决问题的能力。		
教师课前准备	1、布置学习环境。 2、研究和准备教学内容（包括课件，演示项目等）。		
学生课前准备	1、思考什么是计算机语言。 2、准备好教材、课堂记录本等学习用品。		
教学&项目	项目（案例）名称	本次课程项目驱动内容	项目类型
	输出一句话	输出一句问	教师演示-照着做

			候	
	求 x 的正弦值		输入一个数据, 输出其正弦值	教师引导-独立做
	求某学生三年的学费和住宿费		能够写一个数学表达式进行四则运算	教师指导-学着做
知识点			掌握程度 (1-5)	教学策略
二级	三级			
1.C 语言概述	什么是 C 语言		5	<input checked="" type="checkbox"/> 实操学习
	为什么要学习 C 语言		5	<input checked="" type="checkbox"/> 讨论或座谈
2.C 语言结构	C 程序及结构		5	<input checked="" type="checkbox"/> 实操学习
	如何开发 C 程序		5	<input checked="" type="checkbox"/> 实操学习
能力点			掌握程度 (1-5)	教学策略
二级	三级	四级		
程序开发能力	结构化程序设计能力		4	<input checked="" type="checkbox"/> 专题学习
	规范编码能力		5	<input checked="" type="checkbox"/> 专题学习
Dev C++ 平台使用能力	使用平台编码能力		5	加强练习
	调试能力		5	加强练习
阅读、分析、理解 C 语言程序的能力	理解能力		4	探究式项目
素质点			掌握程度 (1-5)	教学策略
二级	三级	四级		
沟通表达	沟通		3	提问、互动
	表达		3	提问、讨论
知识、能力、素质的项目教学活动(任务)历程 (CDIO 代表构思 (Conceive)、设计 (Design)、实施 (Implement)、运行 (Operate))				
活动(任务)一 while 和 do-while				
活动(任务)步骤名称	活动(任务)内容			主要教学策略
问题引入	计算机怎样为人类工作——执行指令			教师讲解学生观察思考

知识讲解	知识点 1: 什么是 C 语言 知识点 2: 为什么学习 C 语言 知识点 3: C 语言结构	启发讲解 讨论归纳
示范操作	(1)教师使用 PPT, 给大家讲解及展示 C 语言的发展过程 (2)教师讲解 C 语言的发展、特点	教师演示, 学生观察思考
知识讲解	知识点 1: C 语言结构 知识点 2: do-while 语句的流程图	启发讲解 讨论归纳
示范操作	教师例题演示: (1) 输出一个字符串 (2) 输入一个实数, 输出它的正弦值	教师例题演示, 学生观察思考
课堂实训	学生上机试写输出字符串例题、求正弦值程序	实践操作 巡视指导 总结分析
活动 (任务) 步骤名称	活动 (任务) 内容	主要教学策略
知识讲解	知识点 1: 如何开发 C 程序	启发讲解 讨论归纳
示范操作	教师演示: (1) 在例一、例二程序编写运行过程中, 所生成的文件 (2) 讲解 C 语言开发的步骤 (3) 讲解可能遇到的错误、查找错误的方法, 调试运行的简单过程	教师例题演示, 学生观察思考
课堂实训	学生仿照修改程序, 查找错误。	实践操作 巡视指导 总结分析
示范操作	教师介绍本单元实例, 分析实例要求, 引导学生完成实例	教师例题演示, 学生观察思考
课堂实训	学生上机实现实例, 试写四则混合运算数学表达式程序	实践操作 巡视指导 总结分析
课堂总结	教师总结本章知识点, 提醒同学学习中注意的问题	整理笔记 布置作业习题讲解
知识拓展	初学者最常问的几个问题	启发讲解
课后作业	1、练习题 2、计算长方形面积	
教学后记	本单元是 C 语言程序设计课程的导入部分, 需要引领学生对 C 语言发展有宏观认识, 并且初步了解 C 语言结构和初学者的注意事项。	

单元名称	单元 2 C 语言基础
授课类型	<input type="checkbox"/> A 类纯理论课 <input checked="" type="checkbox"/> B 类理论+实践 <input type="checkbox"/> C 类纯实践课
学时	理论 4、实验 4
课型	新授
课程思政	在南北朝时期，祖冲之在没有任何先进科学仪器的情况下，将圆周率精确到小数点后七位，即 3.1415926，这一成就领先当时西方将近一千年左右，展示了中国古代数学家的高超智慧和计算能力。 在编程过程中要追求精益求精的工匠精神、创新精神和探索欲望，敢于挑战自我、超越自我。
教学重点及难点 学生学习条件 分析	1、起点分析 (1) 具备使用 DevC++编译器进行程序编写、运行能力 (2) 具备计算机基本操作能力、具备软件安装与卸载能力。 2、重点分析 (3) 常量与变量的含义 (4) 符号常量的使用方法 (5) 标识符的命名规则； (6) 各种常量、变量的的定义和使用方法； (7) 各种运算符的运算规则，各种表达式计算过程； (8) 数据类型转换 3、难点分析 (1) 变量的存储和使用 (2) 运算符与表达式的使用方法
教学方法手段	1、教学方法：采用讲授法、课堂讨论法、练习法、实训法、演示法。 2、学习方法：合作学习，实训学习，自主学习。 3、教学手段：多媒体，项目演示。
教学资源	1、学校资源：多媒体教室、实训教室。 2、文献资源：参考教材，课程大纲，课外读物，实训要求。 3、网络资源：学习通、二维码扫描视频。
能力指标及课程教学目标	1、具备使用利用运算符对各类数据编写正确表达式，解决实际问题的能力 2、具备利用结构化程序设计思想解决实际问题的能力
单元教学目标	2、知识目标 (1) 理解常量与变量的含义 (2) 掌握符号常量的使用方法 (3) 掌握标识符的命名规则； (4) 理解和掌握各种常量、变量的的定义和使用方法； (5) 掌握各种运算符的运算规则，各种表达式计算过程； 2、能力目标 (1) 具备熟练运用运算符、应用不同类型数据编写正确表达式，解决实际问题的能力 3、素养目标 (1) 建立初步编程解决问题的思想。 (2) 具有独立解决问题的能力。
教师课前准备	1、布置学习环境。

	2、研究和准备教学内容（包括课件，演示项目等）。			
学生课前准备	1、思考现实生活中有哪些类型的数据 2、准备好教材、课堂记录本等学习用品。			
教学&项目	项目（案例）名称	本次课程项目驱动内容	项目类型	
	使用符号常量计算圆形面积和周长	定义一个符号常量表示圆周率，计算圆形面积和周长。	教师演示-照着做	
	整型变量的使用	声明几个整型变量，赋值运算后输出结果	教师引导-独立做	
	转义字符的使用	输出多行包含转义字符的字符串，分析输出结果	教师指导-学着做	
	字符型变量和整型变量的运算	字符型数据加减运算、字符型数据和整型数据的相互赋值	教师指导-学着做	
	除法运算符、求余数运算符的应用	应用除法运算符、求余数运算符进行运算	教师演示-照着做	
	赋值运算的类型转换	将不同类型数据相互赋值，观察分析输出结果	教师演示-照着做	
	自动类型转换和强制类型转换	讲解自动类型转换和强制类型转换实例，观察分析输出结果	教师指导-学着做	
	几个特殊运算符的应用	复合赋值运算符、自增自减运算符、逗号运算符、条件运算符、求字节运算符的应用举例	教师演示-照着做	
	实例分析 1：计算多种商品的总价格 实例分析 2：分析表达式的数据类型	定义符号常量、普通变量，计算多个数据的四则运算结果，分析四则混合运算的结果类型	教师指导-学着做	
知识点				
	二级	三级	掌握程度 (1-5)	教学策略
1.数据类型			4	<input checked="" type="checkbox"/> 讨论或讲授
2.常量与变量	常量和符号常量		5	
	变量		5	<input checked="" type="checkbox"/> 实操学习
3.整型数据	整型常量		5	<input checked="" type="checkbox"/> 实操学习
	整型变量		5	<input checked="" type="checkbox"/> 实操学习
4.实型数据	实型常量		5	<input checked="" type="checkbox"/> 实操学习
	实型变量		5	<input checked="" type="checkbox"/> 实操学习
5.字符型数据	字符常量		5	<input checked="" type="checkbox"/> 实操学习
	字符型变量		5	<input checked="" type="checkbox"/> 实操学习
	字符串常量		5	<input checked="" type="checkbox"/> 实操学习
6.运算符与表	算术运算符和算术表达式		5	<input checked="" type="checkbox"/> 实操学习

达式	赋值运算符和赋值表达式		5	<input checked="" type="checkbox"/> 实操学习
	数据类型转换		5	<input checked="" type="checkbox"/> 实操学习
	几个特殊的运算符		5	<input checked="" type="checkbox"/> 实操学习
能力点			掌握程度 (1-5)	教学策略
二级	三级	四级		
程序开发能力	结构化程序设计能力		4	<input checked="" type="checkbox"/> 专题学习
	规范编码能力		5	<input checked="" type="checkbox"/> 专题学习
Dev C++ 平台使用能力	使用平台编码能力		5	加强练习
	调试能力		5	加强练习
阅读、分析、理解 C 语言程序的能力	理解能力		4	探究式项目
素质点			掌握程度 (1-5)	教学策略
二级	三级	四级		
沟通表达	沟通		3	提问、互动
	表达		3	提问、讨论
知识、能力、素质的项目教学活动（任务）历程 (CDIO 代表构思 (Conceive)、设计 (Design)、实施 (Implement)、运行 (Operate))				
活动（任务）一 数据类型、常量和变量基础				
活动（任务）步骤名称	活动（任务）内容			主要教学策略
问题引入	常见信息管理系统中有哪些类型的数据			教师讲解 学生观察思考
知识讲解	知识点 1: 数据类型分类 知识点 2: 常量与变量 知识点 3: 标识符的命名规则 知识点 4: 符号常量的使用			启发讲解 讨论归纳
示范操作	使用符号常量求圆形面积和周长			教师演示, 学生观察思考
课堂实训	学生上机仿照老师示例, 使用符号常量编写程序			实践操作 巡视指导 总结分析
活动（任务）步骤名称	活动（任务）内容			主要教学策略

知识讲解	知识点 1: 整型常量 知识点 2: 整型变量	启发讲解 讨论归纳
示范操作	教师演示: (4) 整型变量的定义与使用	教师例题演示, 学生观察思考
课堂实训	学生仿照练习整型变量的练习程序, 查找错误。	实践操作 巡视指导 总结分析
知识讲解	知识点 1: 实型常量 知识点 2: 实型变量	启发讲解 讨论归纳
示范操作	教师演示实型变量的使用, 讲解实型数据超出计算范围引发的问题	教师例题演示, 学生观察思考
课堂实训	学生仿照练习实型变量的例题程序, 查找错误。	实践操作 巡视指导 总结分析
活动(任务) 步骤名称	活动(任务)内容	主要教学策略
知识讲解	知识点 1: 字符型常量 知识点 2: 字符型变量 知识点 3: 字符型数据和整型数据的转换和运算	启发讲解 讨论归纳
示范操作	教师演示转义字符的示例和字符型变量赋值给整型变量的示例	教师例题演示, 学生观察思考
课堂实训	学生仿照练习大小写字母相互转换的练习程序, 查找错误。	实践操作 巡视指导 总结分析
知识讲解	知识点: 字符串常量	启发讲解 讨论归纳
课堂实训	(1) 学生完成练一练 (2) 做习题: 写出程序的运行结果	实践操作 巡视指导 总结分析
活动(任务)四 运算符与表达式以及数据类型转换		
活动(任务) 步骤名称	活动(任务)内容	主要教学策略
知识讲解	知识点 1: 算术运算符	启发讲解 讨论归纳
示范操作	教师演示除法运算符、求余运算符的应用	教师例题演示, 学生观察思考
课堂实训	学生仿照例题实训	实践操作 巡视指导 总结分析
知识讲解	知识点 2: 算术表达式 知识点 3: 赋值运算符和赋值表达式 知识点 4: 数据类型转换 知识点 5: 赋值运算符类型转换	启发讲解 讨论归纳

	知识点 6: 自动类型转换 知识点 7: 强制类型转换	
示范操作	教师演示赋值运算类型的转换、数据类型自动转换、强制类型转换的示例	教师例题演示, 学生观察思考
课堂实训	学生仿照例题实训, 查找错误	实践操作 巡视指导 总结分析
活动(任务)五 几个特殊的运算符		
活动(任务) 步骤名称	活动(任务)内容	主要教学策略
知识讲解	知识点 1: 复合赋值运算符 知识点 2: 自增运算符和自减运算符 知识点 3: 逗号运算符 知识点 4: 求字节运算符	启发讲解 讨论归纳
示范操作	教师演示各种运算符的使用示例	教师例题演示, 学生观察思考
课堂实训	学生仿照例题实训	实践操作 巡视指导 总结分析
示范操作	教师演示赋值运算类型的转换	教师例题演示, 学生观察思考
课堂实训	学生仿照例题实训, 查找错误	实践操作 巡视指导 总结分析
活动(任务)六 实例分析与实现		
活动(任务) 步骤名称	活动(任务)内容	主要教学策略
课堂实训	学生完成本单元实训: (1) 计算多种商品的总价格 (2) 分析表达式的数据类型	实践操作 巡视指导 总结分析
课堂总结	教师总结本章知识点, 提醒同学学习中注意的问题	整理笔记 布置作业习题讲解
课后作业	1、本单元同步练习题	
教学后记	本单元内容是 C 语言的只是部分, 虽然不难理解, 但对于刚接触编程的学生来说, 会觉得内容复杂, 千头万绪。本单元的教学应把握难度适中, 使学生有信心完成后面的学习。	

单元名称	单元3 顺序结构
授课类型	<input type="checkbox"/> A 类纯理论课 <input checked="" type="checkbox"/> B 类理论+实践 <input type="checkbox"/> C 类纯实践课
学时	理论 2、实验 2
课型	新授
课程思政	<p>程序执行是顺序的、不可跳跃的，引导学生认识到在学习和生活中，也需要脚踏实地，按照既定的计划和步骤逐步前进，不能急于求成或跳跃式前进，养成踏实、稳重的学习态度和生活方式。</p> <p>顺序结构需要严格遵守语法规则和逻辑顺序，任何错误或遗漏都可能导致程序无法正常运行。引导学生认识到在社会生活中也需要遵守各种规则和制度，如法律法规、校规校纪等，增强学生的规则意识和法律意识，使其成为遵纪守法的公民。</p>
教学重点及难点 学生学习条件 分析	<p>1、起点分析</p> <p>(1) 具备使用 DEVC++编译器进行程序编写、运行能力</p> <p>(2) 具备计算机基本操作能力、具备软件安装与卸载能力。</p> <p>(3) 学习了 C 语言基础知识</p> <p>2、重点分析</p> <p>(1) 算法的概念</p> <p>(2) 画流程图的方法</p> <p>(3) 结构化程序设计的原则</p> <p>(4) C 语言的语句分类</p> <p>(5) 格式化数据输入与输出方法</p> <p>3、难点分析</p> <p>(1) 格式化数据的输入与输出</p>
教学方法手段	<p>1、教学方法：采用讲授法、课堂讨论法、练习法、实训法、演示法。</p> <p>2、学习方法：合作学习，实训学习，自主学习。</p> <p>3、教学手段：多媒体，项目演示。</p>
教学资源	<p>1、学校资源：多媒体教室、实训教室。</p> <p>2、文献资源：参考教材，课程大纲，课外读物，实训要求。</p> <p>3、网络资源：学习通、二维码扫描视频。</p>
能力指标及课程教学目标	<p>1、能够正确画流程图</p> <p>2、能够正确输入输出各种格式数据</p>
单元教学目标	<p>3、知识目标</p> <p>(1) 理解算法的概念</p> <p>(2) 掌握画流程图的方法</p> <p>(3) 理解结构化程序设计的原则</p> <p>(4) C 语言的语句分类</p> <p>(5) 掌握格式化数据输入与输出方法</p> <p>2、能力目标</p> <p>(2) 能够熟练运用数据输入输出函数进行数据的读入与输出</p> <p>(3) 具备利用结构化程序设计方法分析问题的能力</p> <p>3、素养目标</p> <p>(1) 建立结构化程序设计的思想。</p>

	(2) 具有独立解决问题的能力。			
教师课前准备	1、布置学习环境。 2、研究和准备教学内容（包括课件，演示项目等）。			
学生课前准备	1、思考依赖于计算机软件程序完成的工作，大体经过哪几个步骤 2、准备好教材、课堂记录本等学习用品。			
教学&项目	项目（案例）名称	本次课程项目驱动内容	项目类型	
	用流程图表示算法	画出求整数的阶乘算法的流程图	教师演示-照着做	
	字符的输入与输出	字符输入与输出函数的应用实例	教师引导-独立做	
	格式化数据输入与输出	格式化输入函数与输出函数的应用实例	教师演示-照着做	
	字符型变量和整型变量的运算	字符型数据加减运算、字符型数据和整型数据的相互赋值	教师引导-独立做	
	数据输入输出项目练习	ATM机取款实例 鸡兔同笼问题	教师演示-照着做	
知识点		掌握程度	教学策略	
二级	三级	(1-5)		
顺序结构	算法与结构化程序设计		3	<input checked="" type="checkbox"/> 讨论或座谈
	C语言语句		5	<input checked="" type="checkbox"/> 讨论或座谈
	字符输入与输出		5	<input checked="" type="checkbox"/> 实操学习
	格式化输入与输出		5	<input checked="" type="checkbox"/> 实操学习
能力点			掌握程度	教学策略
二级	三级	四级	(1-5)	
程序开发能力	结构化程序设计能力		4	<input checked="" type="checkbox"/> 专题学习
	规范编码能力		5	<input checked="" type="checkbox"/> 专题学习
Dev C++ 平台使用能力	使用平台编码能力		5	加强练习
	调试能力		5	加强练习
	算法设计		3	项目练习
素质点			掌握程度	教学策略
二级	三级	四级	(1-5)	
团队协作	合作意识		3	分组、互评

	团队认识		4	分组、互评
知识、能力、素质的项目教学活动（任务）历程 （CDIO 代表构思（Conceive）、设计（Design）、实施（Implement）、运行（Operate））				
活动（任务）一 算法与结构化程序设计				
活动（任务） 步骤名称	活动（任务）内容			主要教学策略
问题引入	计算机软件通常分几步完成任务。			学生观察思考，教师讲解演示
知识讲解	知识点 1：算法的概念 知识点 2：算法的特性 知识点 3：算法与流程图 知识点 4：结构化程序设计原则			启发讲解 讨论归纳
示范操作	教师例题演示： （1） 写出算法求 5! 的值，探讨如何改进算法。 （2） 用流程图表示求 5! 的算法 （3） 用 N-S 流程图表示 5! 的算法			教师例题演示，学生观察思考
课堂实训	（1） 学生写算法求 $1+2+\dots+100$ （2） 学生画出求 $1+2+\dots+100$ 的流程图 （3） 学生画出求 $1+2+\dots+100$ 的 N-S 流程图			实践操作 巡视指导 总结分析
活动（任务）二 C 语言语句				
活动（任务） 步骤名称	活动（任务）内容			主要教学策略
知识讲解	知识点 5：C 语言程序的构成 知识点 6：C 语句的分类 知识点 7：字符输入与输出函数 知识点 8：格式化输入与输出函数			启发讲解 讨论归纳
示范操作	教师例题演示： （1） 输出单个字符 （2） 输入单个字符 （3） 使用 printf 函数输出数据的示例 （4） 使用 scanf 函数输入数据的示例			教师例题演示，学生观察思考
课堂实训	（1） 学生仿写例题输入输出单个字符 （2） 学生仿写例题输入输出格式化数据 为提高学生们对中华文化的學習热情，某班级特举办“学习强国”答题竞赛。请编程模拟答题过程。 例如：“魏晋时期的（ ）因主持编绘《禹贡地域图》和提出“制图六体”而被称为地图学家。 （3） 鸡兔同笼趣味问题的编程实现 （4） 求圆周长、面积、球体表面积、球体体积、圆柱体体积 （5） 输入小写字母，把小写字母转换成大写字母输出。			实践操作 巡视指导

知识拓展	C 程序代码编写规范	启发讲解 讨论归纳
课堂总结	<ul style="list-style-type: none"> (1) 算法的概念 (2) 画流程图的方法 (3) 结构化程序设计的原则 (4) C 语言的语句分类 (5) 格式化数据输入与输出方法 	总结分析 整理笔记
课后作业	<ul style="list-style-type: none"> 1、本单元同步习题 2、编程题目 	
教学后记	本章最重要一点就是让学生通过本章的学习逐渐建立起模块化程序设计的思想，把大问题分解成小问题进行解决。	

单元名称	单元 4 选择结构
授课类型	<input type="checkbox"/> A 类纯理论课 <input checked="" type="checkbox"/> B 类理论+实践 <input type="checkbox"/> C 类纯实践课
学时	理论 4、实验 4
课型	新授
课程思政	选择结构中的条件判断需要严格遵守语法规则和逻辑规则，任何错误都可能导致程序运行失败。强调逻辑思维能力和批判性思维能力，培养学生在面对复杂问题时能够保持清醒的头脑和正确的方向。 另外，引导学生认识到在社会生活中也需要遵守各种规则和制度，如法律法规、校规校纪等，增强学生的规则意识和法治精神，增强学生的社会责任感和道德感，能够坚守正确的价值观和道德标准。
教学重点及难点 学生学习条件 分析	1、起点分析 (1) 具备使用 DEVC++编译器进行程序编写、运行能力 (2) 具备计算机基本操作能力、具备软件安装与卸载能力。 (3) 学习了 C 语言基础知识、顺序结构、输入输出数据的方法 2、重点分析 (1) 条件判断表达式 (2) if 选择语句 (3) switch 语句 (4) 选择结构程序设计方法 3、难点分析 (1) if 语句和 switch 语句
教学方法手段	1、教学方法：采用讲授法、课堂讨论法、练习法、实训法、演示法。 2、学习方法：合作学习，实训学习，自主学习。 3、教学手段：多媒体，项目演示。
教学资源	1、学校资源：多媒体教室、实训教室。 2、文献资源：参考教材，课程大纲，课外读物，实训要求。 3、网络资源：学习通、二维码扫描视频。
能力指标及课程教学目标	3、能够正确使用 if 语句的不同形式能够正确使用 switch 语句控制程序流程
单元教学目标	4、知识目标 (1) 掌握关系、逻辑运算符的运算规则，能用 C 语言表达式解决实际问题。 (2) 掌握 if 语句的三种基本形式、了解 if 语句的嵌套； (3) 掌握 switch...case 语句的使用方法。 (4) 掌握选择结构程序设计方法。 2、能力目标 (1) 具备熟练运用选择结构解决实际问题的能力 3、素养目标 (1) 建立结构化程序设计的思想。 (2) 具有独立解决问题的能力。
教师课前准备	1、布置学习环境。 2、研究和准备教学内容（包括课件，演示项目等）。
学生课前准备	1、思考根据判断条件选择执行的操作时，如何描述判断条件

2、准备好教材、课堂记录本等学习用品。						
教学&项目	项目（案例）名称		本次课程项目驱动内容	项目类型		
	if 选择语句		关系运算符和关系表达式	教师演示-照着做		
	输入两个整数进行排序		输入 a、b 的值，比较二者大小，从小到大输出	教师引导-独立做		
	判断一个整数是正数还是负数		根据输入数据的正负不同，选择不同的输出信息	教师演示-照着做		
	根据输入 x 的值，求出对应 y 的值		根据输入 x 的值的多种情况，选择 y 对应的值输出	教师引导-独立做		
	划分学生考试成绩等级		判断学生成绩区域，选择对应等级输出	教师演示-照着做		
输入数字 1-7，输出对应日期		根据输入数字，用 switch 语句分多种情况输出对应日期	教师演示-照着做			
计算商场打折促销时的商品价格。		多分支 if 语句和 switch 语句的使用	教师引导-独立做			
知识点			掌握程度	教学策略		
二级	三级		(1-5)			
选择结构	条件判断表达式		5	<input checked="" type="checkbox"/> 讨论或座谈		
	if 选择语句		5	<input checked="" type="checkbox"/> 讨论或座谈		
	switch 语句		5	<input checked="" type="checkbox"/> 实操学习		
能力点				掌握程度	教学策略	
二级	三级	四级	(1-5)			
程序开发能力	结构化程序设计能力		4	<input checked="" type="checkbox"/> 专题学习		
	规范编码能力		5	<input checked="" type="checkbox"/> 专题学习		
Dev C++ 平台使用能力	使用平台编码能力		5	加强练习		
	调试能力		5	加强练习		
	算法设计		3	项目练习		
素质点				掌握程度	教学策略	
二级	三级	四级	(1-5)			
团队协作	合作意识		3	分组、互评		
	团队认识		4	分组、互评		

知识、能力、素质的项目教学活动（任务）历程 (CDIO 代表构思 (Conceive)、设计 (Design)、实施 (Implement)、运行 (Operate))		
活动（任务）一 条件判断表达式		
活动（任务） 步骤名称	活动（任务）内容	主要教学策略
问题引入	面临选择的时候，人们需要根据已有条件寻求各种解决方案	学生观察思考，教师讲解演示
知识讲解	知识点 1：条件判断表达式 知识点 2：关系运算符和关系表达式 知识点 3：逻辑运算符和逻辑表达式	启发讲解 讨论归纳
示范操作	教师例题演示： (4) 计算关系表达式的值 (5) 计算逻辑表达式的值	教师例题演示，学生观察思考
课堂实训	(4) 学生求出关系表达式和逻辑表达式的值	实践操作 巡视指导 总结分析
活动（任务） 步骤名称	活动（任务）内容	主要教学策略
知识讲解	知识点 4：单分支 if 语句 知识点 5：双分支 if 语句 知识点 6：多分支 if 语句 知识点 7：条件运算符 知识点 8：switch 语句	启发讲解 讨论归纳
示范操作	教师例题演示： (5) 输入两个整数，输出两个数中的最大值 (6) 判断某年是否是闰年 (7) 根据输入 x 的值，输出对应 y 的值 (8) 对学生成绩进行等级划分 (9) 输入 1-7 的值，输出对应的日期	教师例题演示，学生观察思考
课堂实训	(6) 学生仿写例题实现单分支结构程序 (7) 学生仿写例题实现双分支结构程序 (8) 学生仿写例题实现多分支结构程序 (9) 商场收款程序：实现不同商品单价、购买数量、不同折扣标准情况下，商品总价。	实践操作 巡视指导
知识拓展	程序中的语法错误和逻辑错误调试	启发讲解 讨论归纳
课堂总结	(6) 关系运算符和逻辑运算符 (7) if 语句 (8) switch 语句	总结分析 整理笔记
课后作业	1、本单元同步习题 2、编程题目	

教学后记	本单元最重要就是让学生通过本章的学习，学会如何根据已知条件编写关系表达式和逻辑表达式，正确运用分支结构完成程序设计。教学过程中需要注意引导学生理顺清晰的编程思路。
------	---

单元名称	单元 5 循环结构		
授课类型	<input type="checkbox"/> A 类纯理论课	<input checked="" type="checkbox"/> B 类理论+实践	<input type="checkbox"/> C 类纯实践课
学时	理论 6、实验 6		
课型	新授		
课程思政	循环结构中的重复执行过程象征着坚持不懈的毅力，引导学生理解循环结构的工作原理，并强调在面对困难和挑战时，要像程序中的循环结构一样，要有严谨的态度，并持之以恒、不断尝试，坚持不懈，直到达到目标。培养学生的坚韧不拔、勇于面对挑战的精神，使他们能够在未来的学习和工作中不断克服困难，取得更大的成就。		
教学重点及难点 学生学习条件分析	1、起点分析 (1) 具备 C 语言程序设计基础知识，顺序结构、选择结构。 (2) 具备计算机基本操作能力、具备软件安装与卸载能力。 2、重点分析 (1) while、do-while 和 for 基本循环语句的使用方法。 (2) 循环嵌套。 (3) break 和 continue 语句。 3、难点分析 (1) 循环嵌套。 (2) break 和 continue 语句的区别。		
教学方法手段	1、教学方法：采用讲授法、课堂讨论法、练习法、实训法、演示法。 2、学习方法：合作学习，实训学习，自主学习。 3、教学手段：多媒体，项目演示。		
教学资源	1、学校资源：多媒体教室、实训教室。 2、文献资源：参考教材，课程大纲，课外读物，实训要求。 3、网络资源：学习通、二维码扫描视频。		
能力指标及课程教学目标	1、能够使用循环语言打印出乘法口诀表。 2、能够使用循环语言解决循环问题。		
单元教学目标	1、知识目标 (1) 掌握 while、do-while 和 for 语句的结构。 (2) 掌握 break 和 continue 语句的特点。 (3) 掌握循环嵌套的结构。 2、能力目标 (3) 能够使用三种循环语句编写程序。 (4) 能够使用 break 和 continue 语句。 3、素养目标 (1) 建立使用循环语句解决问题的思想。 (2) 具有独立解决问题的能力。		
教师课前准备	1、布置学习环境。 2、研究和准备教学内容（包括课件，演示项目等）。		
学生课前准备	1、思考那些现象是循环现象。 2、准备好教材、课堂记录本等学习用品。		
教学&项目	项目（案例）名称	本次课程项目驱动内容	项目类型

	统计小组考试成绩		统计并输出成绩	教师演示-照着做
	找素数		输出 100—200 之间的素数	教师引导-独立做
	打印出乘法口诀表		能够在屏幕上输出乘法口诀表	教师指导-学着做
知识点			掌握程度 (1-5)	教学策略
二级	三级			
1、循环结构	while 语句		5	<input checked="" type="checkbox"/> 实操学习
	do-while 语句		5	<input checked="" type="checkbox"/> 讨论或座谈
	for 语句		5	<input checked="" type="checkbox"/> 实操学习
	嵌套的循环		5	<input checked="" type="checkbox"/> 实操学习
能力点			掌握程度 (1-5)	教学策略
二级	三级	四级		
程序开发能力	结构化程序设计能力		4	<input checked="" type="checkbox"/> 专题学习
	规范编码能力		5	<input checked="" type="checkbox"/> 专题学习
Dev C++ 平台使用能力	使用平台编码能力		5	加强练习
	调试能力		5	加强练习
阅读、分析、理解 C 语言程序的能力	理解能力		4	探究式项目
素质点			掌握程度 (1-5)	教学策略
二级	三级	四级		
沟通表达	沟通		3	提问、互动
	表达		3	提问、讨论
知识、能力、素质的项目教学活动（任务）历程 (CDIO 代表构思 (Conceive)、设计 (Design)、实施 (Implement)、运行 (Operate))				
活动（任务）一 while 和 do-while				
活动（任务）步骤名称	活动（任务）内容			主要教学策略
问题引入	举例说明什么情况下适合使用循环。例如：求 1 到 100 的累加和			教师讲解学生观察思考

知识讲解	知识点 1: while 语句的执行过程 知识点 2: while 语句的流程图	启发讲解 讨论归纳
示范操作	教师例题演示: (1) 使用 while 语句, 输出 50 个 “*”。 (2) 用 while 语句求 n! (3) 用 while 语句统计一个小组学生考试成绩	教师例题演示, 学生观察思考
知识讲解	知识点 1: do-while 语句的执行过程 知识点 2: do-while 语句的流程图	启发讲解 讨论归纳
示范操作	教师例题演示: (1) 用 do-while 语句统计一个小组学生考试成绩 (2) 使用 do-while 语句, 输出 50 个 “*”。	教师例题演示, 学生观察思考
课堂实训	学生上机试写 while 和 do-while 例题程序	实践操作 巡视指导 总结分析
活动(任务)二 For 循环		
活动(任务)步骤名称	活动(任务)内容	主要教学策略
知识讲解	知识点 1: for 语句的执行过程 知识点 2: for 语句的流程图	启发讲解 讨论归纳
示范操作	教师例题演示: (5) 用 for 语句统计一个小组学生考试成绩 (6) 用 for 语句计算象棋棋盘存放大米的数量。	教师例题演示, 学生观察思考
课堂实训	学生上机仿照例题, 试写 for 语句例题程序	实践操作 巡视指导 总结分析
知识讲解	知识点 1: for 语句的使用注意事项 知识点 2: 循环嵌套	启发讲解 讨论归纳
示范操作	教师例题演示: 1. 在屏幕上输出如下图形 ***** ***** ***** 2. 输出乘法九九口诀表。	教师例题演示, 学生观察思考
课堂实训	学生上机仿照例题, 试写循环语句例题程序	实践操作 巡视指导 总结分析
活动(任务)三 嵌套循环		
活动(任务)步骤名称	活动(任务)内容	主要教学策略
示范操作	知识点: break 语句和 continue 语句	教师例题演示, 学生观察思考

知识讲解	<p>教师例题演示：</p> <ol style="list-style-type: none"> 1. 使用 for 循环语句输出 50 个 “*”， break 控制数量。 2. 使用 for 循环语句输出 50 个 “*”，使用 continue 控制数量。 	启发讲解 讨论归纳
示范操作	while 语句、 do-while 语句和 for 语句的使用	教师例题演示，学生观察思考
课堂实训	算法的时间复杂度	分析程序 实践操作 巡视指导
知识讲解	<p>本章三种循环语句的特点</p> <p>break 语句与 continue 语句的区别使用注意事项</p>	启发讲解 讨论归纳
课堂总结	<p>总结</p> <p>本章三种循环语句的特点</p> <p>break 语句与 continue 语句的区别使用注意事项</p>	整理笔记 布置作业习题讲解
课后作业	<ol style="list-style-type: none"> 1、练习题 2、计算 1 到 20 的阶乘之和 	
教学后记	<p>本单元最重要就是让学生通过本章的学习，学会如何根据已知条件编写关系表达式和逻辑表达式，正确运用分支结构完成程序设计。教学过程中需要注意引导学生理顺清晰的编程思路。</p>	

单元名称	单元 6 数组		
授课类型	<input type="checkbox"/> A 类纯理论课	<input checked="" type="checkbox"/> B 类理论+实践	<input type="checkbox"/> C 类纯实践课
学时	理论 4、实验 4		
课型	新授		
课程思政	中国南宋时期数学家杨辉所著的《详解九章算法》(1261 年)中提出了杨辉三角,在欧洲,这一规律由法国数学家帕斯卡(1623-1662)在 1654 年独立发现,但比杨辉晚了 393 年。杨辉三角在数学、计算机科学、物理学等领域都有广泛的应用。		
教学重点及难点 学生学习条件分析	1、起点分析 (1) 具备 C 语言程序设计基础知识,顺序结构、选择结构,循环结构。 (2) 具备计算机基本操作能力、具备软件安装与卸载能力。 2、重点分析 (1) 一维数组的定义、初始化及引用方法。 (2) 数组的输入、输出、排序。 3、难点分析 (1) 二维数组的定义、初始化及引用方法。 (2) 数组的输入、输出、排序。		
教学方法手段	1、教学方法:采用讲授法、课堂讨论法、练习法、实训法、演示法。 2、学习方法:合作学习,实训学习,自主学习。 3、教学手段:多媒体,项目演示。		
教学资源	1、学校资源:多媒体教室、实训教室。 2、文献资源:参考教材,课程大纲,课外读物,实训要求。 3、网络资源:学习通、二维码扫描视频。		
能力指标及课程教学目标	1、能够使用数组进行排序。 2、能够使用数组解决实际问题。		
单元教学目标	1、知识目标 (1) 掌握数组的定义、初始化及引用方法。 (2) 掌握数组的输入、输出、排序。 2、能力目标 (1) 能够使用数组进行排序。 (2) 能够使用数组解决实际问题。 3、素养目标 (1) 建立使用数组解决问题的思想。 (2) 具有独立解决问题的能力。		
教师课前准备	1、布置学习环境。 2、研究和准备教学内容(包括课件,演示项目等)。		
学生课前准备	1、查询数组的概念。 2、准备好教材、课堂记录本等学习用品。		
教学&项目	项目(案例)名称	本次课程项目驱动内容	项目类型
	排序	数据排序	教师演示-照着做
	字符串处理	文字处理	教师引导-独立做

知识点		掌握程度 (1-5)	教学策略	
二级	三级			
6、数组	一维数组	5	<input checked="" type="checkbox"/> 实操学习	
	二维数组	5	<input checked="" type="checkbox"/> 讨论或座谈	
	字符数组和字符串	5	<input checked="" type="checkbox"/> 实操学习	
能力点			掌握程度 (1-5)	教学策略
二级	三级	四级		
程序开发能力	结构化程序设计能力		4	<input checked="" type="checkbox"/> 专题学习
	规范编码能力		5	<input checked="" type="checkbox"/> 专题学习
Dev C++ 平台使用能力	使用平台编码能力		5	加强练习
	调试能力		5	加强练习
问题抽象	数据结构描述		3	项目练习
	算法设计		3	项目练习
素质点			掌握程度 (1-5)	教学策略
二级	三级	四级		
团队协作	合作意识		3	分组、互评
	团队认识		4	分组、互评
知识、能力、素质的项目教学活动（任务）历程 (CDIO 代表构思 (Conceive)、设计 (Design)、实施 (Implement)、运行 (Operate))				
活动（任务）一 排序				
活动（任务） 步骤名称	活动（任务）内容			主要教学策略
问题引入	思考:多个相同数据类型变量的定义与使用			教师讲解演示, 学生观察思考
知识讲解	知识点 1: 一维数组的定义、初始化、引用。 知识点 2: 一维数组作函数参数。			启发讲解 讨论归纳
示范操作	教师例题演示: 1、一维数组的赋值与显示 2、用数组来处理求 Fibonacci 数列问题。 3、用选择法对 10 个整数进行由大到小排序。			教师例题演示, 学生观察思考

课堂实训	(1) 用冒泡法对 10 个数进行由小到大排序。	实践操作 巡视指导 总结分析
活动（任务）二 一维数组的应用		
活动（任务） 步骤名称	活动（任务）内容	主要教学策略
知识讲解	一维数组的应用。	启发讲解 讨论归纳
示范操作	教师例题演示 1: 判别一个整数数组中各元素的值，若大于 0 则输出该值，若小于或等于 0 则输出 0 值。 教师例题演示 2: 从键盘输入 10 个整型数据，找出其中的最小值并输出。	教师例题演示，学生观察思考
课堂实训	(1) 数组 a 中存放了一个学生 7 门课程的成绩，求平均成绩。 (2) 向有序数组中插入一个数，保持原顺序不变，将新数组输出。	实践操作 巡视指导
课堂总结	(1) 一维数组的定义、初始化、引用中的注意事项； (2) 一维数组元素、一维数组名作函数参数两者的区别及应用； (3) 一维数组的应用——排序算法。	总结分析 整理笔记
活动（任务）三 二维数组应用		
活动（任务） 步骤名称	活动（任务）内容	主要教学策略
知识讲解	知识点 1: 二维数组的定义、初始化、引用 知识点 2: 二维数组的应用	启发讲解 讨论归纳
示范操作	教师例题演示： 1: 二维数组按行分段赋值 2: 一个学习小组有 5 个人，每个人有三门课的考试成绩。求全组分科的平均成绩和各科总平均成绩。 3: 求二维数组（5 行 5 列）中最大元素值及其行列号	教师例题演示，学生观察思考
课堂实训	(1) 编写程序，实现读取学生的学号和成绩信息，并输出。	实践操作 巡视指导
课堂总结	(1) 使用二维数组中的注意事项； (2) 二维数组中常见操作的实现； (3) 多维数组与二维数组的类比。	总结分析 整理笔记
活动（任务）三 二维数组应用		
活动（任务） 步骤名称	活动（任务）内容	主要教学策略

知识讲解	<p>知识点 1: 字符串与字符数组</p> <p>知识点 2: 字符数组的输入与输出</p> <p>知识点 3: 字符串处理函数</p>	启发讲解 讨论归纳
示范操作	<p>教师例题演示:</p> <p>1: 字符数组的引用举例</p> <p>2: 字符数组的输入、输出举例</p> <p>3: 逐个输入 10 个大写字母, 依次打印其所对应的小写字母。</p> <p>4: 我国自古以来是一个多民族国家。中华人民共和国成立后, 经中央人民政府调查与统计正式确认的民族共有 56 个。汉族是我国的主体民族, 约占全国人口总数的 91.11%, 其他还有 55 个民族, 约占 8.89%。汉族和 55 个少数民族共同组成伟大的中华民族。我国, 是一个以汉族为主体、56 个民族共同组成的统一的多民族国家。请编程输出我国 56 个民族名称。</p>	教师例题演示, 学生观察思考
课堂实训	<p>(1) 为了保证信息的安全, 大多数系统都含有用户登录模块。只有输入正确的用户名和密码之后才能进行相应的操作。编写程序实现用户登录功能。</p> <p>(2) 《孔子家语》中有一则名言“言必诚信, 行必忠正”。诚信是一个人的立身之本, 也是一个集体、一个民族、一个国家的生存之基。现编程实现输入社会主义核心价值观 24 字, 并输出。</p>	实践操作 巡视指导 总结分析
知识拓展	算法的空间复杂度	启发讲解 讨论归纳
课堂总结	<p>本章主要内容回顾</p> <p>一维数组的定义与引用</p> <p>二维数组的定义与引用</p> <p>字符串和字符数组</p> <p>常用字符串处理函数</p>	整理笔记 布置作业习题讲解
课后作业	<p>1、杨辉三角输出</p> <p>2、本章教材及习题内容</p>	
教学后记	本单元内容相对独立, 单独讨论数组的使用。该单元的知识在应用时需要综合运用前面各章学习的知识, 重点在于使学生理解数组在内存中的存储方式, 以及运用循环语句时操作的数组的哪个数据。	
单元名称	单元 7 函数	
授课类型	<input type="checkbox"/> A 类纯理论课 <input checked="" type="checkbox"/> B 类理论+实践 <input type="checkbox"/> C 类纯实践课	

学时	理论 2、实验 2		
课型	新授		
课程思政	函数定义时需要遵循一定的语法规则，如函数名、参数列表、返回类型等都需要明确指定。在调用函数时，也需要按照规定的参数类型和数量传递实参。引导学生认识到在编程中需要严格遵守规则，任何违反规则的行为都可能导致程序错误。培养学生的规则意识、逻辑思维、团队协作和创新精神，保持高度的责任心和敬业精神。		
教学重点及难点 学生学习条件分析	1、起点分析 (1) 具备 C 语言程序设计基础知识，顺序结构、选择结构、循环结构，数组。 (2) 具备计算机基本操作能力、具备软件安装与卸载能力。 2、重点分析 (1) 函数的定义。 (2) 函数的调用。 (3) 变量的作用域。 3、难点分析 (1) 递归函数。		
教学方法手段	1、教学方法：采用讲授法、课堂讨论法、练习法、实训法、演示法。 2、学习方法：合作学习，实训学习，自主学习。 3、教学手段：多媒体，项目演示。		
教学资源	1、学校资源：多媒体教室、实训教室。 2、文献资源：参考教材，课程大纲，课外读物，实训要求。 3、网络资源：学习通、二维码扫描视频。		
能力指标及课程教学目标	1、能够使用函数求两数之和。2、能够使用函数计算 1~N 之和。		
单元教学目标	1、知识目标 (1) 掌握函数的定义形式。 (2) 掌握函数的调用。 (3) 掌握变量的作用域。 2、能力目标 (1) 能够使用函数求两数之和。 (2) 能够使用函数计算 1~N 之和。 3、素养目标 (1) 建立模块化程序设计的思想。 (2) 具有独立解决问题的能力。		
教师课前准备	1、布置学习环境。 2、研究和准备教学内容（包括课件，演示项目等）。		
学生课前准备	1、查询函数的概念。 2、准备好教材、课堂记录本等学习用品。		
教学&项目	项目（案例）名称	本次课程项目驱动内容	项目类型
	求两数之和	使用函数	教师演示-照着做
	求 1~n 之和	递归函数	教师引导-独立做

知识点		掌握程度 (1-5)	编号			
二级	三级					
7、函数	函数定义	5	<input checked="" type="checkbox"/> 实操学习			
	函数调用	5	<input checked="" type="checkbox"/> 讨论或座谈			
	函数的嵌套调用	5	<input checked="" type="checkbox"/> 实操学习			
	函数的递归调用	5	<input checked="" type="checkbox"/> 实操学习			
	函数的声明	5	<input checked="" type="checkbox"/> 讨论或座谈			
	变量的作用域和存储类别	5	<input checked="" type="checkbox"/> 讨论或座谈			
能力点			掌握程度 (1-5)	编号	呼应的能力 指标	教学策 略
二级	三级	四级				
程序开发能力	结构化程序 设计能力		4	<input checked="" type="checkbox"/> 专题学习		
	规范编码能 力		5	<input checked="" type="checkbox"/> 专题学习		
Dev C++ 平台 使用能力	使用平台编 码能力		5	加强练习		
	调试能力		5	加强练习		
	算法设计		3	项目练习		
素质点			掌握程度 (1-5)	编号		
二级	三级	四级				
团队协作	合作意识		3	分组、互评		
	团队认识		4	分组、互评		
知识、能力、素质的项目教学活动（任务）历程 (CDIO 代表构思 (Conceive)、设计 (Design)、实施 (Implement)、运行 (Operate))						
活动（任务）一 函数的定义与调用						
活动（任务） 步骤名称	活动（任务）内容				主要教学策略	
问题引入	怎样实现程序的独立性				学生观察思考，教 师讲解演示	
知识讲解	知识点 1：函数在结构化程序设计中的意义； 知识点 2：函数的定义； 知识点 3：函数的调用方法；				启发讲解 讨论归纳	
示范操作	教师例题演示： 1、定义一个函数，实现两个整数的求和运算，并将 结果返回。 2、定义一无参函数，实现两个整数的求和运算。				教师例题演示，学 生观察思考	

	3、定义一个无参函数，实现菜单提示功能。 4、有参函数的调用，调用求和运算的函数。	
课堂实训	(1) 编写一个函数，实现两个数的交换，并在主函数中调用。 (2) 编写一个简易计算器程序，能够进行加、减、乘、除计算功能。	实践操作 巡视指导 总结分析
活动（任务） 步骤名称	活动（任务）内容	主要教学策略
知识讲解	知识点 4：函数原型与函数定义； 知识点 5：函数的嵌套调用和递归调用； 知识点 6：局部变量与全局变量的定义。	启发讲解 讨论归纳
示范操作	教师例题演示： 1、计算 1~n 之间整数之和； 2、有 5 个人坐在一起，问第五个人多少岁？他说比第 4 个人大 2 岁。问第 4 个人岁数，他说比第 3 个人大 2 岁。问第三个人，又说比第 2 人大两岁。问第 2 个人，说比第 1 个人大两岁。最后问第 1 个人，他说是 10 岁。请问第五个人多大？	教师例题演示，学生观察思考
课堂实训	(1) 用递归的方法设计一个函数，求 N! (2) 使用函数的递归调用求解 Ficonacci 数列。 (3) 设计一个函数，使用全局变量，实现三个数的排序。	实践操作 巡视指导
知识拓展	模块化程序设计	启发讲解 讨论归纳
课堂总结	(1) 函数定义的一般形式。 (2) 无参函数和有参函数的调用过程，函数原型以及函数的返回值。 (3) 函数递归调用的过程。 (4) 全局变量和局部变量的生存期和作用域。	总结分析 整理笔记
课后作业	1、定义判断素数函数 2、Fibonacci 数列	
教学后记	本章最重要一点就是让学生通过本章的学习逐渐建立起模块化程序设计的思想，把大问题分解成小问题进行解决。	
单元名称	单元 8 指针	
授课类型	<input type="checkbox"/> A 类纯理论课 <input checked="" type="checkbox"/> B 类理论+实践 <input type="checkbox"/> C 类纯实践课	

学时	理论 2、实验 2		
课型	新授		
课程思政	指针不当操作可能导致程序崩溃、数据泄露等安全问题。指针在网络安全领域中的重要性，培养学生的信息安全意识。		
教学重点及难点 学生学习条件分析	1、起点分析 (1) 具备 C 语言程序设计基础知识，顺序结构、选择结构、循环结构，数组、函数。 (2) 具备计算机基本操作能力、具备软件安装与卸载能力。 2、重点分析 (1) 指针的定义和使用。 (2) 指向数组的指针。 3、难点分析 (1) 用指针变量作为函数参数。 (2) 返回指针的函数。		
教学方法手段	1、教学方法：采用讲授法、课堂讨论法、练习法、实训法、演示法。 2、学习方法：合作学习，实训学习，自主学习。 3、教学手段：多媒体，项目演示。		
教学资源	1、学校资源：多媒体教室、实训教室。 2、文献资源：参考教材，课程大纲，课外读物，实训要求。 3、网络资源：学习通、二维码扫描视频。		
能力指标及课程教学目标	1、能够使用指针进行排序。 2、能够使用指针输出字符串。		
单元教学目标	1、知识目标 (1) 掌握地址和指针的概念。 (2) 掌握指针变量的定义和使用。 (3) 掌握指向数组的指针的使用方法。 (4) 掌握指针变量作为函数参加的使用方法。 (5) 了解返回指针的函数。 2、能力目标 (1) 能够使用指针进行排序。 (2) 能够使用指针输出字符串。 3、素养目标 (1) 建立规范的编程行为能力。 (2) 具有独立解决问题的能力。		
教师课前准备	1、布置学习环境。 2、研究和准备教学内容（包括课件，演示项目等）。		
学生课前准备	1、查询函数的概念。 2、准备好教材、课堂记录本等学习用品。		
教学&项目	项目（案例）名称	本次课程项目驱动内容	项目类型
	排序	使用指针排序	教师演示-照着做
	字符串处理	使用指针输出字符串	教师引导-独立做
	求最大值	使用指针两个数进行交换	教师引导-独立做
	知识点	掌握程	教学策略

二级	三级		度 (1-5)	
8、指针	地址和指针		5	<input checked="" type="checkbox"/> 实操学习
	指针变量		5	<input checked="" type="checkbox"/> 讨论或座谈
	指向数组的指针		5	<input checked="" type="checkbox"/> 实操学习
	用指针变量作为函数参数		4	<input checked="" type="checkbox"/> 实操学习
	返回指针值的函数		4	<input checked="" type="checkbox"/> 讨论或座谈
能力点			掌握程 度 (1-5)	教学策略
二级	三级	四级		
程序开发能力	结构化程序 设计能力		4	<input checked="" type="checkbox"/> 专题学习
	规范编码能 力		5	<input checked="" type="checkbox"/> 专题学习
Dev C++ 平台 使用能力	使用平台编 码能力		5	加强练习
	调试能力		5	加强练习
	算法设计		3	项目练习
素质点			掌握程 度 (1-5)	教学策略
二级	三级	四级		
团队协作	合作意识		3	分组、互评
	团队认识		4	分组、互评
知识、能力、素质的项目教学活动（任务）历程 (CDIO 代表构思 (Conceive)、设计 (Design)、实施 (Implement)、运行 (Operate))				
活动（任务）一 两数排序				
活动（任务） 步骤名称	活动（任务）内容			主要教学策略
问题引入	计算机存储器中数据的存放方式			教师讲解演示，学 生观察思考
知识讲解	知识点 1: 指针的定义 知识点 2: 指针对所指向数据的引用			启发讲解 讨论归纳
示范操作	教师例题演示，学生观察思考， (1)通过指针变量访问整型变量 (2)取地址符号与取值符号是互为逆运算的。 (3)利用指针对两个数进行排序			教师例题演示，学 生上机练习
上机实训	学生仿照例题上机练习			实践操作 巡视指导 总结分析
活动（任务）二 学生成绩计算				
活动（任务）	活动（任务）内容			主要教学策略

步骤名称		
知识讲解	知识点 1: 指向一维数组的指针定义 知识点 2: 数组指针的运算 知识点 3: 指针对数组元素的引用	教师讲解演示, 学生观察思考
示范操作	教师例题演示, 学生观察思考 (1) 指针变量的运算 (2) 显示指针变量的当前值 (3) 输入 5 名学生的成绩, 计算平均分, 输出高于平均分的成绩, 使用指针完成数组元素的操作。	启发讲解 讨论归纳
上机实训	学生仿照例题上机练习	教师例题演示, 学生上机练习
活动(任务)三 字符串处理		
活动(任务)步骤名称	活动(任务)内容	主要教学策略
知识讲解	知识点: 指向字符串的指针	启发讲解 讨论归纳
示范操作	教师例题演示: 使用字符型指针复制字符串	教师例题演示, 学生观察思考
上机实训	学生仿照例题上机练习	实践操作 巡视指导 总结分析
活动(任务)四 使用指针指向二维数组		
活动(任务)步骤名称	活动(任务)内容	主要教学策略
知识讲解	知识点 1: 指向二维数组的指针 知识点 1: 指向二维数组的列指针	启发讲解 讨论归纳
示范操作	教师例题演示: 使用列指针输出二维数组元素	教师例题演示, 学生观察思考
上机实训	学生仿照例题上机练习	实践操作 巡视指导 总结分析
知识讲解	知识点 1: 指向二维数组的行指针	启发讲解 讨论归纳
示范操作	教师例题演示: (1) 用行指针输出二维数组元素。 (2) 二维数组元素的不同表示方法	教师例题演示, 学生观察思考
上机实训	学生仿照例题上机练习	实践操作 巡视指导 总结分析
活动(任务)五 使用指针变量作函数参数		
活动(任务)	活动(任务)内容	主要教学策略

步骤名称		
知识讲解	知识点 1: 指针变量作为函数参数 知识点 2: 返回指针的函数	启发讲解 讨论归纳
示范操作	教师例题演示: 指针变量做形参, 交换函数实参变量的值。	教师例题演示, 学生观察思考
上机实训	学生仿照例题上机练习	实践操作 巡视指导 总结分析
知识拓展	引导学生自学 main 函数的形式参数	启发讲解
课堂总结	本章主要内容回顾 <ul style="list-style-type: none"> ▶ 指针变量的使用。 ▶ 指针变量的引用。 ▶ 指针可以指向一维数组、二维数组 ▶ 指针可以作为参数进行函数调用, 函数也可以返回指针类型数据 	整理笔记 布置作业习题讲解
课后作业	1、将字符串的中数字与字母分离 2、教材及习题	
教学后记	指针是一个比较难理解的内容, 学习这部分内容一定要使用比较简单的例子引入, 引起学生的学习兴趣。	

单元名称	单元 11 综合项目实训
授课类型	<input type="checkbox"/> A 类纯理论课 <input checked="" type="checkbox"/> B 类理论+实践 <input type="checkbox"/> C 类纯实践课
学时	理论 4、实验 4
课型	新授
教学目标及要求	掌握用 C 语言编写综合项目
教学重点及难点	C 语言程序设计基础知识，顺序结构、选择结构、循环结构，数组、函数、指针、结构体。
教学方法手段	1、教学方法：采用讲授法、课堂讨论法、练习法、实训法、演示法。 2、学习方法：合作学习，实训学习，自主学习。 3、教学手段：多媒体，项目演示。
教学资源	1、学校资源：多媒体教室、实训教室。 2、文献资源：参考教材，课程大纲，课外读物，实训要求。 3、网络资源：学习通、二维码扫描视频。
本单元代码	<p style="text-align: center;">单元 11</p> <p>【例 11.1】编写程序生成 5 个随机数（伪随机数）。</p> <pre>#include<stdio.h> #include<stdlib.h> int main(){ int n=5; printf("在这个编译环境中生成 5 个 0-%d 的随机数。 \n",RAND_MAX); do{ printf("第%d 个随机数: %d\n",6-n,rand()); n--; }while(n>0); return 0; }</pre> <p>【例 11.2】为例 11.1 加入时间种子，生成真正的随机数。</p> <pre>#include<stdio.h> #include<stdlib.h> #include<time.h> int main(){ int n=5; printf("在这个编译环境中生成 5 个 0-%d 的随机数。 \n",RAND_MAX); srand(time(NULL)); do{ printf("第%d 个随机数: %d\n",6-n,rand()); n--; }while(n>0); return 0; }</pre> <p>【例 11.3】猜数游戏：计算机随机出 0~100 之间的一个整数由玩家来猜，玩家猜中游戏结束。</p> <pre>#include<stdio.h> #include<stdlib.h> #include<time.h></pre>

```

int main() {
    int n=0,x,ans;
    int flag=1;
    printf("请猜一个 0-100 的整数。 \n");
    srand(time(NULL));
    ans=rand()%101;
    do{
        printf("是多少呢? ");
        scanf("%x",&x);
        n++;
        if(x>ans)
            printf("\a 这个数大了。 \n");
        if(x<ans)
            printf("\a 这个数小了。 \n");
        if(x==ans) {
            flag=0;
            printf("\a 恭喜你猜中了, 总共猜了%d 次。 \n",n);
        }while(flag);
    }return 0;
}

```

【例 11.4】 每次判断胜负的猜拳游戏实现。

```

#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main() {
    int human;
    int comp;
    int judge;
    int retry;
    srand(time(NULL));
    printf("猜拳游戏开始!! \n");
    do{
        comp=rand()%3;    //计算机随机产生一个 0~2 的整数, 0-石头, 1-剪刀,
2-布

        printf("\a\n 石头剪刀布...0-石头, 1-剪刀, 2-布: ");
        scanf("%d",&human);
        printf("我出");
        switch(comp) {
            case 0:puts("石头");break;
            case 1:puts("剪刀");break;
            case 2:puts("布");break;
        }
        printf("\n");
        judge=(human-comp+3)%3;    //判断胜负
        switch(judge){
            case 0:puts("平局。");break;
            case 1:puts("你输了。");break;
            case 2:puts("你赢了。");break;
        }
        printf("再来一次吗? 0-否, 1-是");
        scanf("%d",&retry);
    }while(retry==1);
    return 0;
}

```

【例 11.5】

```

#include<stdio.h>
#include<time.h>
#include<stdlib.h>
int human;
int comp;
int win no;
int lose no;
int draw no;
char *hd[]={"石头","剪刀","布"};
/*---初始处理---*/
void initialize(void) {
    win no=0;
}

```

```

lose no=0;
draw no=0;
srand(time(NULL));
printf("猜拳游戏开始!! \n");
}
/*---运行猜拳游戏(读取/生成手势)---*/
void jyanken(void){
int i;
comp=rand()%3;
do{
printf("\n\a石头剪刀布...");
for(i=0;i<3;i++)
printf("(%d)%s",i,hd[i]);
printf(":");
scanf("%d",&human);
}while(human<0||human>2);
}
/*---更新胜利/失败/平局次数---*/
void count_no(int result){
switch(result){
case 0:draw no++;break;
case 1:lose no++;break;
case 2:win no++;break;
}
}
/*---更显示判断结果---*/
void disp_result(int result){
switch(result){
case 0:puts("平局。");break;
case 1:puts("你输了。");break;
case 2:puts("你赢了。");break;
}
}
//主程序
int main(){
int judge;
initialize();
do{
jyanken();
printf("我出%s, 你出%s. \n",hd[comp],hd[human]);
judge=(human-comp+3)%3;
count_no(judge);
disp_result(judge);
}while(win no<3&&lose no<3);
printf("win no==3?"\n你赢了. \n":"\n我赢了. \n");
printf("%d胜%d负%d平\n",win no,lose no,draw no);
return 0;
}

```

【例 11.6】 随机生成由 4 个不同数字组成的数字串。

```

#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main(){
int x[4];
int i,j;
srand(time(NULL));
for(i=0;i<4;i++){
do{
x[i]=rand()%10;
for(j=0;j<i;j++)
if(x[i]==x[j]) break;
}while(j<i);
}
for(i=0;i<4;i++)
printf("%d",x[i]);
return 0;
}

```

【例 11.7】

```
int check(char s[]){
    int i,j;
    if(strlen(s)!=4) return 1;
    for(i=0;i<4;i++){
        if(s[i]<'0' || s[i]>'9')
            return 2;
        for(j=0;j<i;j++){
            if(s[j]==s[i])
                return 3;
        }
    }
    return 0;
}
```

【例 11.8】编写一个函数，判断比较正确答案和玩家输入的数字字符串，返回

hit 和 blow 的值。

```
void judge(char s[],int no[],int *hit,int *blow){
    int i,j;
    *hit=*blow=0;
    for(i=0;i<4;i++){
        for(j=0;j<4;j++){
            if(s[i]=='0'+no[j]) //0~9 数字整型转换成字符串
                if(i==j)
                    (*hit)++;
            else
                (*blow)++;
        }
    }
}
```

【例 11.9】珠玑妙算的程序。

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>
#include<time.h>
/*---生成不同的 4 个数字组合并存入数组 x---*/
void make4digits(int x[]){
    int i,j,val;
    for(i=0;i<4;i++){
        do{
            val=rand()%10;
            for(j=0;j<i;j++){
                if(val==x[j])
                    break;
            }while(j<i);
            x[i]=val;
        }
    }
}
/*---检查已输入的字符串 s 的有效性---*/
int check(char s[]){
    int i,j;
    if(strlen(s)!=4)
        return 1; //长度不为 4,返回 1
    for(i=0;i<4;i++){
        if(!isdigit(s[i]))
            return 2;
        for(j=0;j<i;j++){
            if(s[i]==s[j])
                return 3;
        }
    }
    return 0;
}
/*---hit 和 blow 的判断---*/
```

```

void judge(char s[],int no[],int *hit,int *blow){
    int i,j;
    *hit=*blow=0;
    for(i=0;i<4;i++){
        for(j=0;j<4;j++){
            if(s[i]=='0'+no[j])
                if(i==j) (*hit)++;
                else (*blow)++;
        }
    }
}
/*---显示判断结果---*/
void print_result(int snum,int spos){
    if(spos==4)
        printf("回答正确! ");
    else if(snum==0)
        printf("这些数字里没有答案数字。");
    else{
        printf("这些数字里包括%d个答案数字。",snum);
        if(spos==0)
            printf("但是数字的位置都不一致。\\n");
        else
            printf("其中有%d个数字的位置是一致的。\\n",spos);
    }
    putchar('\\n');
}

int main(){
    int try_no=0;
    int chk;
    int hit,blow,no[4];
    char buff[10];
    clock_t start,end;
    srand(time(NULL));
    puts("■来玩珠玑妙算吧。");
    puts("■请猜 4 个数字。");
    puts("■其中不包含相同的数字。");
    puts("■请像 4307 这样连续输入数字。");
    puts("■不能输入空格字符。\\n");
    make4digits(no);
    start=clock();
    do{
        do{
            printf("请输入: ");
            scanf("%s",buff);
            chk=check(buff);
            switch(chk){
                case 1:puts("\\a 请确保输入 4 个数字。");break;
                case 2:puts("\\a 请不要输入数字以外的字符。");break;
                case 3:puts("\\a 请不要输入相同的数字。");break;
            }
        }while(chk!=0);
        try_no++;
        judge(buff,no,&hit,&blow);
        print_result(hit+blow,hit);
    }while(hit<4);
    end=clock();
    printf(" 用 了 %d 次 。 \\n 用 时 %.1f 秒 。
\\n",try_no,(double)(end-start)/CLOCKS_PER_SEC);
    return 0;
}

```

【例 11.10】编写显示系统当前日期和时间的程序。

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

```

```

#include <windows.h>
//定义日期和时间结构体
typedef struct {
    int year;
    int month;
    int day;
    int hour;
    int minute;
    int second;
    int week;
}myDate;
//判断是否是闰年
int leap(int year) {
    if (year%4 == 0 && year%100 !=0 || year%400 == 0){
        return 1;
    }
    return 0;
}

int main() {
    int days[13] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    char * weekday[] = {"星期日", "星期一", "星期二", "星期三", "星期四", "星期五", "星期六"};
    time t t = time(NULL);
    struct tm *cur = localtime(&t);
    myDate Date = {cur->tm_year+1900, cur->tm_mon + 1, cur->tm_mday, cur->tm_hour, cur->tm_min, cur->tm_sec, cur->tm_wday};
    printf("\n%4d 年%02d 月%02d 日 [%s]%02d 时%02d 分%02d 秒\r", Date.year, Date.month, Date.day, weekday[Date.week], Date.hour, Date.minute, Date.second);
    while(1) {
        Sleep(1000);
        days[2] = (leap(Date.year) ? 28:29);
        Date.second++;
        if ( Date.second == 60) {
            Date.second = 0;
            Date.minute++;
        }
        if (Date.minute == 60){
            Date.minute = 0;
            Date.hour++;
        }
        if (Date.hour == 24) {
            Date.hour = 0;
            Date.week = (++Date.week)%7;
            Date.day++;
        }
        if (Date.day == (days[Date.month] + 1)) {
            Date.day = 1;
            Date.month++;
        }
        if (Date.month == 13) {
            Date.month = 1;
            Date.year++;
        }
        printf("%4d 年%02d 月%02d 日 [%s]%02d 时%02d 分%02d 秒\r", Date.year, Date.month, Date.day, weekday[Date.week], Date.hour, Date.minute, Date.second);
    }
    return 0;
}

```

【例 11.11】模拟七段数码管动态显示日期和时间。

```
#include <stdio.h>
```

```

#include <stdlib.h>
#include <time.h>
#include <windows.h>
#define MAX DIGHTS 20

int leap(int year);
void clear dight array(void);
void process dights array(int dight, int position);
void print dights array(void);
const int segments[11][7] = {
    {1, 1, 1, 1, 1, 1, 0}, /* 0 */
    {0, 1, 1, 0, 0, 0, 0}, /* 1 */
    {1, 1, 0, 1, 1, 0, 1}, /* 2 */
    {1, 1, 1, 1, 0, 0, 1}, /* 3 */
    {0, 1, 1, 0, 0, 1, 1}, /* 4 */
    {1, 0, 1, 1, 0, 1, 1}, /* 5 */
    {1, 0, 1, 1, 1, 1, 1}, /* 6 */
    {1, 1, 1, 0, 0, 0, 0}, /* 7 */
    {1, 1, 1, 1, 1, 1, 1}, /* 8 */
    {1, 1, 1, 1, 0, 1, 1}, /* 9 */
    {0, 0, 0, 0, 0, 0, 1} /* - */
};
char dights[3][MAX DIGHTS * 4];
typedef struct
{
    int year;
    int month;
    int day;
    int hour;
    int minute;
    int second;
    int week;
}myDate;

int main()
{
    int days[13] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
31};
    time_t t = time(NULL);
    struct tm *cur = localtime(&t);
    myDate Date = {cur->tm year+1900, cur->tm mon+1, cur->tm mday,
cur->tm hour,
cur->tm min, cur->tm sec, cur->tm wday};
    while(1)
    {
        days[2] = (leap(Date.year) ? 28:29);
        Date.second++;
        if ( Date.second == 60) {
            Date.second = 0;
            Date.minute++;
        }
        if (Date.minute == 60) {
            Date.minute = 0;
            Date.hour++;
        }
        if (Date.hour == 24) {
            Date.hour = 0;
            Date.day++;
        }
        if (Date.day == (days[Date.month] + 1)) {
            Date.day = 1;
            Date.month++;
        }
        if (Date.month == 13) {
            Date.month = 1;
            Date.year++;
        }
    }
    clear dight array();
    //显示日期: 年-月-日

```

```

process dights array(Date.year/1000, 0);
process dights array(Date.year/100%10, 1);
process dights array(Date.year/10%10, 2);
process dights array(Date.year%10, 3);
process dights array(10, 4);
process dights array(Date.month/10, 5);
process dights array(Date.month%10, 6);
process dights array(10, 7);
process dights array(Date.day/10, 8);
process dights array(Date.day%10, 9);
//显示时间: 时-分-秒
process dights array(Date.hour/10, 11);
process dights array(Date.hour%10, 12);
process dights array(10, 13);
process dights array(Date.minute/10, 14);
process dights array(Date.minute%10, 15);
process dights array(10, 16);
process dights array(Date.second/10, 17);
process dights array(Date.second%10, 18);
print dights array();
Sleep(1000); //延时1秒
system("cls"); //清屏
}
return 0;
}

int leap(int year) //判断是否是闰年
{
    if (year%4 == 0 && year%100 !=0 || year%400 == 0)
    {
        return 1;
    }
    return 0;
}

void clear_dight_array(void) {
    memset(dights, ' ', sizeof dights); //内存初始化
}

/*显示一个数字及位置*/
void process_dights_array(int dight, int position) {
    int n = position * 4;
    if (segments[dight][0])
        dights[0][n + 1] = ' ';
    if (segments[dight][1])
        dights[1][n + 2] = '|';
    if (segments[dight][2])
        dights[2][n + 2] = '|';
    if (segments[dight][3])
        dights[2][n + 1] = ' ';
    if (segments[dight][4])
        dights[2][n] = '|';
    if (segments[dight][5])
        dights[1][n] = '|';
    if (segments[dight][6])
        dights[1][n + 1] = ' ';
}

void print_dights_array(void) {
    int i, j;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < MAX_DIGITS * 4; j++) {
            putchar(dights[i][j]);
        }
        putchar('\n');
    }
}
}

```

课后作业	本单元同步习题
教学后记	本单元考查学生的综合能力，要多实践才能提高编程能力，适应各种工作岗位需求。