CERACU

全国高等学校计算机教育研究会团体标准

T/CERACU XXXXX—XXXX

计算机核心课程规范
计算机程序设计

Specification for Core Courses of Computer Science

Computer Programming

（征求意见稿）

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

全国高等学校计算机教育研究会 发布

目 次

目次 I
前言 II
计算机核心课程规范 计算机程序设计 3
1 范围 3
2 概述 3
3 课程目标与总体要求 4
3.1 课程目标与教学要求 4
3.2 考核要求 4
4 基础知识点及要求 4
5 面向对象程程序设计扩展知识点及要求 7
6 泛型程序设计扩展知识点及要求 9
7 算法设计扩展知识点及要求 11
8 事件驱动的程序设计模式扩展知识点及要求 12
附 录 B （资料性附录）“C++语言程序设计”课程知识体系与教学要求..... 14
C.1 课程说明 14
C.2 教学要求 14
附 录 C （资料性附录） 本科 C#语言程序设计课程教学大纲对标准覆盖情况案例 18
C.1 课程说明 18
C.2 教学要求 18
C.1 课程说明 23
C.2 C 语言程序设计课程教学内容..... 23
附 录 D （资料性附录）“C 语言程序设计”（线上线下混合）课程教学要求（案例） 32
D.1 课程说明 32
D.2 教学要求 32
附 录 E XXXX ..... 错误!未定义书签。
E.1.1 XXXX 39

前　言

本标准按照GB/T 1.1－2009给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由全国高等学校计算机教育研究会（CERACU）提出并归口。

本标准起草单位：。

本标准主要起草人：

计算机核心课程规范
计算机程序设计

1 范围

本标准给出了高等学校计算机程序设计课程的知识点集合、知识体系结构，规定了课程的教学要求。

本标准适用于高等学校计算机程序设计课程、教材、教学过程和教学评价。

2 概述

本标准主要包括与具体计算机程序设计语言无关的基础知识点及要求、算法设计的扩展知识点及要求、面向对象程序设计的扩展知识点及要求、泛型程序设计的扩展知识点及要求、事件驱动的程序设计扩展知识点及要求。

知识体系按照分层定义，结构层级不超过三级，每个知识点可以容纳下级知识点，也可以是原子知识点。能力要求给出了解、掌握、综合应用三个层次的具体要求。学时要求是应至少达到的学时下限，不设上限，包括总学时、理论学时（包括听课、自学等各种形式）、实践学时（包括实验、项目实训等各种形式），“其他”可根据不同语言知识点的需要进行扩展。

知识点属性如表1所示。

表1 知识点属性

属性项	数据类型	取值范围	可选（O）/必备（M）	说明
标识符	字符串	长度不超过20	M	在本标准中唯一标识知识点
名称	字符串	长度不超过40	M	应采用行业内共识的知识点名称
说明	字符串	长度不超过400	O	说明知识点内涵。非原子知识点O，原子知识点M
能力要求	容器类型	包含了解、掌握、综合应用三个子项	M	
了解	字符串	长度不超过1000	O	说明某知识点“了解”要求的具体含义
掌握	字符串	长度不超过1000	O	说明某知识点“掌握”要求的具体含义
综合应用	字符串	长度不超过1000	O	说明某知识点“综合应用”要求的具体含义
学时要求	容器类型	包括理论、实践、其他三个子项	O/M	顶层知识点为M；非顶层知识点的为O
总学时	整数	大于0	O/M	O/M规则同“学时要求”。此项为生成项，等于理论、实践、其他三项之和

理论	整数	大于或等于0	O/M	O/M规则同“学时要求”。理论、实践、其他至少一项不为0，且三项总和与总学时相等
实践	整数	大于或等于0	O/M	O/M规则同“学时要求”。理论、实践、其他至少一项不为0，且三项总和与总学时相等
其他	整数	大于或等于0	O/M	O/M规则同“学时要求”。理论、实践、其他至少一项不为0，且三项总和与总学时相等

3 课程目标与总体要求

3.1 课程目标与教学要求

课程目标应服从各个学校、院系的培养方案。本标准不规定具体课程的教学要求，宜从本标准中选择适用的知识点和能力要求、学时要求组成一门具体课程的内容和要求，应包括本标准第7章规定的知识点80%以上，能力要求为“掌握”、“综合应用”的应占所选知识点的80%以上。本标准中未包含的知识点应参照本标准规定的格式进行描述。

3.2 考核要求

课程考核要求应服从各个学校、院系的培养方案和教学实际需求。

考核知识点应覆盖所有被选择作为该课程内容的知识点，考题难度应与该课程所选的各知识点能力要求、学时要求相符。每道题目应明确对应到一个或多个知识点及能力要求。各知识点题目比例应与学时比例一致，含多个知识点的综合题目宜按学习次序靠后的知识点和难度较大的一个或多个知识点计。

考核形式应适合知识点的内容和能力要求。

4 基础知识点及要求

基础知识点及要求如表2所示

表2 基础知识点及要求

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
PROG_FD_01	基础语法		能够看懂只包含基础语法的简单程序	能够应用基础语法编写简单程序		2	1	1	0
PROG_FD_01_01	顺序结构的程序		初步认识最简单的程序	能够仿照例题编写最简单的程序					
PROG_FD_01_02	字符集		知道字符集的概念和定义的语法	能够在程序中正确使用字符集					
PROG_FD_01_03	词法记号		知道词法记号的概念和定义的语法	能够在程序中正确使用词法记号					

PROG_FD_01_04	标准输入/输出	关联到标准设备的输入与输出	能够以默认格式进行标准输入/输出	能够控制标准输入/输出的宽度、精度等简单格式	能够应用语言规定的所有格式控制方式进行标准输入/输出；能够进行标准输入/输出重定向				
PROG_FD_01_05	语句		知道语句的语法	能够正确使用语句					
PROG_FD_02	数据类型		知道数据类型的概念和基本使用	能够按照设计要求定义数据的类型，并使用各种类型的语句	能够针对问题需求选择恰当的数据类型，并正确使用	1	0.5	0.5	0
PROG_FD_02_01	基本数据类型	程序语言语法内置的数据类型	知道基本数据类型的概念和基本使用	能够按照要求使用基本数据类型处理数据	能够根据需求选择使用恰当的数据类型处理数据				
PROG_FD_02_02	类型转换		知道各种数据类型之间的转换规则	能够按要求正确进行数据类型转换	能够通过数据类型转换解决实际问题				
PROG_FD_03	数据的存储与内置运算		知道常量、变量和内置运算的概念和语法	能够按照要求使用常量、变量和内置运算	能够根据需求选择使用常量、变量和各种内置运算符实现各种运算	2	1	1	0
PROG_FD_03_01	常量		知道常量定义和使用的语法	能够按照要求定义和使用常量	能够按照问题需求设计和使用常量				
PROG_FD_03_02	变量		知道变量定义和使用的语法	能够按照要求定义和使用变量	能够按照问题需求设计和使用变量				
PROG_FD_03_03	内置运算		知道各种内置运算符的优先级和结合性	能够按照要求使用内置运算符实现运算	能够按照问题需求使用内置运算符实现运算				
PROG_FD_03_04	表达式		知道表达式的构成语法	能够按要求使用常量、变量和内置运算符构成表达式	能够根据运算需求设计表达式				
PROG_FD_04	流程控制		知道选择结构与循环结构	能够按照要求使用选择与循环结构	能够根据问题需求设计与实现程序的流程控制	4	2	2	0
PROG_FD_04_01	选择语句	语法规定的选择语句和开关语句	知道选择结构的概念和语法	能够根据要求使用合适的选择结构编写程序	能够根据问题需求设计和实现多重选择结构的程序	1.5	1	0.5	0
PROG_FD_04_02	循环语句	语法规定的各种循环语句及其嵌套	知道循环结构的概念和语法	能够根据要求使用合适的循环结构编写程序	能够根据问题需求设计和实现嵌套循环结构的程序	2	1	1	0

PROG_FD_04_03	其他控制语句		知道其他控制语句的语法	能够根据问题要求恰当选择使用		0.5	0.5	0	0
PROG_FD_05	功能模块	表达程序功能的基本单位。例如：函数、方法	知道功能模块的定义和使用方法	能够按照要求定义和使用功能模块	能够根据问题的需求设计和实现模块，并构建模块间的关系	4	2	2	0
PROG_FD_05_01	定义		知道功能模块的定义语法	能够按照要求定义功能模块	能够根据问题的需求设计功能模块并实现	1	0.5	0.5	0
PROG_FD_05_02	调用		知道功能模块的调用语法	能够按照要求调用功能模块	能够根据问题的需求进行功能模块调用，以及模块的嵌套和递归调用	2	1	1	0
PROG_FD_05_03	功能模块间的数据共享	例如函数调用时的数据传递	知道功能模块间数据共享的语法	能够按照要求实现功能模块间数据共享	能够根据问题的需求设计和实现功能模块间的数据共享	1	0.5	0.5	0
PROG_FD_06	数组		知道数组的概念和语法	能够按照要求定义和使用数组	能够根据问题的需求设计和使用数组	4	2	2	0
PROG_FD_07	地址与指针（C/C++）		知道地址类型和指针的概念和语法			3	1.5	1.5	0
PROG_FD_07_01	地址运算		知道地址运算语法	能够按照设计要求使用地址运算	能够根据问题的需求恰当使用地址运算				
PROG_FD_07_02	指针类型数据的定义		能够定义指针类型常量和变量	能够按照设计要求定义和使用指针	能够根据问题的需求恰当使用指针				
PROG_FD_07_03	指针运算	包含指针的算术运算、关系运算和赋值运算	知道指针运算语法	能够按照设计要求使用指针运算	能够根据问题的需求恰当使用指针运算				
PROG_FD_08	引用（C++/Java）		知道引用的语法	能够按照设计要求使用引用	能够根据问题的需求恰当使用引用	1	0.5	0.5	0
PROG_FD_09	字符串		知道字符串存储与表示的语法	能够按照要求使用字符串	能够根据问题的需求定义和使用字符串	2	1	1	0
PROG_FD_10	输入/输出	实现数据的输入/输出，例如：文件操作、数据持久化	知道输入/输出概念和处理方法	能够根据要求实现输入/输出与文件操作	能够理解问题需求，设计和实现合适的输入/输出与文件操作	4	2	2	0
PROG_FD_10_01	输入/输		知道输入/输	能够根据要求	能够理解问题需	1	0.5	0.5	0

	出的基本概念和语法		出的概念、语法和标准库基本操作	实现输入/输出	求，设计和实现合适的输入/输出				
PROG_FD_10_02	文件操作		知道文件处理的语法和标准库基本操作	能够根据要求实现文件操作	能够理解问题需求，设计和实现合适的文件操作	3	1.5	1.5	0
PROG_FD_11	标准库	程序语言标准/规范所规定的库	知道标准库的结构和调用方法	能够根据要求使用标准库中的预定义模块	能够理解问题需求，在解决方案中有效利用标准库中的预定义模块	3	1.5	1.5	0
PROG_FD_11_01	标准库组成	库的主要组成部分	知道标准库的组成结构	能够根据要求选择使用标准库中的相应模块		0.5	0.5	0	0
PROG_FD_11_02	标准库调用	使用标准库的语法	知道语法上如何使用标准库中的预定义模块	能够根据要求在程序中使用标准库中的预定义模块	能够理解问题需求，有效选择和利用标准库中的预定义模块，设计恰当有效的解决方案	2.5	1	1.5	0
PROG_FD_12	程序开发过程与工具环境	程序的编辑、编译链接（或解释）等过程，以及相应的开发环境/工具	知道程序的开发过程，知道至少一种开发环境及其中的工具集	能够使用至少一种开发环境及其中的基本工具集，完成简单程序的开发过程	能够根据需要，选择合适的开发环境，精通多种工具，完成复杂程序的开发和调试	1	0	1	0
PROG_FD_12_01	程序的开发过程	程序的编辑、编译链接（或解释）等过程	知道程序的开发过程	能够完成简单程序的开发和调试	能够完成复杂程序的开发和调试	0.5		0.5	0
PROG_FD_12_02	开发环境/工具		知道至少一种开发环境及其中的工具集	能够使用至少一种开发环境及其中的基本工具集	能够根据需要，选择合适的开发环境，精通多种工具	0.5		0.5	0

5 面向对象程程序设计扩展知识点及要求

面向对象程程序设计扩展知识点及要求如表3所示

表3 面向对象程程序设计扩展知识点及要求

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
PROG_OO_01	类与对象		知道类与对象的概念、语法和用途	能够按照要求定义和使用类和对象	能够对问题进行抽象，设计类结构和定义类，根据需求定义和使用对象	3	1.5	1.5	0
PROG_OO_01_01	类的概念	类定义的语	知道类的	能够按照要求	能够对问题进				

	和定义	法，以及抽象和封装的思想	概念和类定义的语法	定义和使用类	行抽象，设计类结构、定义和使用类				
PROG_OO_01_01_01	类的成员	包括数据成员或属性、函数成员或方法	知道类成员定义的语法	能够按照要求定义和使用类的成员	能够对问题进行抽象，设计、定义和使用成员				
PROG_OO_01_01_02	成员的访问方式和权限	类定义时规定的可访问成员的方式和在不同范围的访问权限	知道访问成员的方式和访问权限定义的语法	能够按照要求定义成员的访问方式和权限	能够对问题进行抽象，设计和定义成员的访问方式和权限				
PROG_OO_01_01_03	函数/方法重载	在同一作用域范围内，定义名字相同，但参数表不同的函数，包括全局函数重载和类成员函数重载	知道函数/方法重载的概念和语法	能够按照要求定义重载函数/方法	能够根据问题需求定义和使用重载函数/方法				
PROG_OO_01_02	对象的概念和定义	对象定义的语法，对象与类的关系，对象在程序中的作用	知道对象的概念、定义和使用	能够按照要求定义和使用对象	能够根据问题需求定义和使用对象				
PROG_OO_01_03	对象的初始化和删除	对象初始化的意义和作用，初始化方法，对象不再使用时被删除的方法及其意义	知道对象可以初始化和删除	能够按照要求对对象进行初始化和删除	能够根据问题需求，设计和实现对象的初始化和删除				
PROG_OO_01_04	对象的复制		知道对象复制的概念和方法	能够按照要求复制对象		0.5	0.25	0,25	0
PROG_OO_01_05	对象在函数或方法间共享	通过类的封装、函数或方法参数和返回值等，在内存中共享对象	知道对象在函数或方法间共享的语法规定	能够按照要求定义共享机制	能够根据问题需求，设计和实现对象在函数或方法间的共享				
PROG_OO_02	类的重用	利用已有的类设计新类	知道类重用的基本概念和语法	能够按照要求定义类的重用关系	能够根据问题需求，设计和实现类的重用	8	4	4	0
PROG_OO_02_01	类的组合	利用部件类的对象作为成员设计组合类	知道类组合的基本概念和语法	能够按照要求定义类的组合关系	能够根据问题需求，设计和实现类的组合	2	1	1	0
PROG_OO_02_01_01	组合类的定义	定义含有对象成员的类	知道组合类定义的语法	能够按照要求实现组合类的定义	能够根据问题需求，设计和实现类的组合				
PROG_OO_02_01_02	组合类对象的初始化和删除	组合类对象初始化和删除时基本类型成员和对象成员的处理（不需要	知道组合类对象初始化时要对成员对象进行初	能够按照要求在类定义时设计和实现基本类型成员和对象成员的初始	能够根据问题需求，设计和实现组合类对象的多种初始化和删除方法				

		考虑初始化或删除的语言除外）	始化	化和删除					
PROG_OO_02_02	类之间的引用	在类成员函数或方法参数表、局部对象等处引用其他类	知道类之间引用的语法	能够按照要求实现类之间的引用关系	能够根据问题需求，设计和实现类之间的引用关系	0.5	0.25	0.25	0
PROG_OO_02_03	类的继承	在继承基类或超类的基础上设计新的类（派生类或子类）	知道类继承的概念、语法和应用	能够按照要求实现类之间的继承关系、对象的初始化和删除	能够根据问题需求，设计和实现类之间的继承关系、对象的初始化和删除	3	1.5	1.5	0
PROG_OO_02_03_01	继承关系的定义	定义类与类之间的继承关系	知道类继承的语法	能够按照设计要求设计和实现类之间的继承关系	能够根据问题需求，设计和实现类之间的继承关系				
PROG_OO_02_03_02	派生类对象的初始化和删除	在对象诞生和消亡时进行初始化和删除（不需要考虑初始化或删除的语言除外）	知道对象诞生时需要初始化、消亡时需要删除	能够按照要求实现对象初始化和删除	能够根据问题需求，设计和实现对象初始化和删除				
PROG_OO_02_04	多态性	包括静态多态性和动态多态性	知道多态性的概念、动态多态性语法和用途	能够按照要求实现具有动态多态性的程序	能够根据问题需求，设计和实现具有动态多态性的程序	2.5	1.25	1.25	0
PROG_OO_02_04_01	多态性的概念和语法	多态性的概念含义、语法	知道多态性的概念和语法	能够按照要求实现具有多态性的程序	能够根据问题需求，设计和实现具有多态性的程序				
PROG_OO_02_04_02	多态性的应用	多态性的应用场合、解决的问题和效果	知道多态性的用途	能够编写程序，实现设计中的多态性意图	能够理解问题需求，通过多态性使解决方案更优				
PROG_OO_03	异常处理	在程序设计语言中用于描述异常与异常处置而用的语言机制	知道异常处理的概念和语法	能够按照要求实现异常处理	能够理解问题需求，通过异常处理机制增强程序的容错能力	2	1	1	0
PROG_OO_03_01	异常处理的基本思想	异常处理的目的、意义和用途	知道异常处理的基本概念和使用	能够理解设计中的异常处理要求	能够理解问题需求，识别对异常处理的需求	0.5	0.5	0	0
PROG_OO_03_02	异常处理的语法	程序设计语言中对异常处理机制的规定	知道异常处理的语法	能够根据设计中的异常处理要求实现异常处理	能够理解问题需求，识别对异常处理的需求并实现异常处理	1.5	0.5	1	0

6 泛型程序设计扩展知识点及要求

表4 泛型程序设计扩展基础知识点及要求

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总学时	理论	实践	其他
PROG_GN_01	泛型程序设计		知道泛型程序设计的概念、语法和使用	能够按照要求定义和使用泛型	能够根据问题需求，设计和实现泛型程序	2	2	1	0
PROG_GN_01_01	函数模板的概念和定义	函数模板的语法，类型参数化的思想	知道函数模板的概念和函数模板定义的语法	能够按照要求定义和使用函数模板	能够根据问题需求，设计和实现函数模板				
PROG_GN_01_02	类模板的概念和定义	类模板定义的语法，对不同数据类型的类进行抽象	知道类模板的概念和定义、使用的语法	能够按照要求定义和使用类模板	能够根据问题需求，设计和实现类模板				
PROG_GN_01_03	模板的实例化	实例化的意义；隐式实例化与显式实例化的区别	知道模板需要实例化后才能使用	能够区分隐式实例化与显式实例化	能够根据问题需求，在多文件下组织模板，并实例化				
PROG_GN_01_04	模板的特殊实现	模板的特化、类模板的偏特化、函数模板的重载的定义与语法	知道可以实现模板特化，了解其作用与意义	能够根据要求实现类模板的偏特化、函数模板的重载	能够根据问题需求，设计和实现特殊模板				
PROG_GN_02	泛型应用					3	2	1	0
PROG_GN_02_01	容器的概念与定义	容器的定义和使用的语法；提供存储和组织元素的方法	知道容器是一组元素的对象；知道容器的分类与作用	能够按照要求定义和使用容器	能够根据问题需求，选择和使用合适的容器				
PROG_GN_02_02	迭代器的概念与定义	迭代器的定义和使用的语法；提供访问元素	知道迭代器提供访问元素的方法；知道迭代	能够按照要求定义和使用迭代器	能够根据问题需求，选择和使用合适的迭代器				

		方法	器的分类与作用						

7 算法设计扩展知识点及要求

表5 算法设计扩展知识点及要求

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	灵活应用	总学时	理论	实践	其他
PROG_AL_01	穷举	根据实际问题建立适合穷举的数学模型,并通过编程实现	知道穷举法的基本思想、特点和优劣	能够根据要求，使用穷举数学模型求解问题	能够根据问题需求，建立穷举数学模型并优化实现	2	1	1	0
PROG_AL_02	递推	根据实际问题建立适合递推的数学模型,并通过编程实现	知道递推算法的基本思想、特点和优劣	能够根据要求，使用递推数学模型求解问题	能够根据问题需求，建立递推数学模型并优化实现	2	1	1	0
PROG_AL_03	迭代	根据实际问题建立适合迭代的数学模型,并通过编程实现	知道迭代算法的基本思想、特点和优缺点	能够根据要求，使用迭代数学模型求解问题	能够根据问题需求，建立迭代数学模型并实现	2	1	1	0
PROG_AL_04	递归	通过函数自调用的方式,不断缩小问题规模直到问题可以直接得到解决	知道递归算法的基本思想,理解递归过程中问题规模的变化	能够根据要求，使用递归求解问题	能够根据问题需求，建立递归数学模型并实现	2	1	1	0

PROG_AL_05	排序	常见排序算法：冒泡排序、选择排序、插入排序等	知道排序算法的基本思想	能够根据要求，选择合适的排序算法解决问题	能够根据问题需求，设计并实现排序算法	2	1	1	0
PROG_AL_06	贪心	通过一系列局部最优决策来计算整个问题的最优解	知道贪心算法的求解过程和特点	能够根据要求，实现贪心算法	能够根据问题需求，判断问题是否适用贪心算法，能用贪心算法求解问题	2	1	1	0

8 事件驱动的程序设计模式扩展知识点及要求

表6 事件驱动的程序设计扩展基础知识点及要求

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总学时	理论	实践	其他
PROG_EV_01	事件驱动程序设计	事件的基本概念、事件源、事件对象、事件处理器、监听与广播模式、事件响应机制的实现	知道事件的基本概念和使用	能够根据要求，实现事件源、事件对象、事件处理器的基本应用	能够根据问题需求，设计和实现监听与广播模式、事件响应机制	2	2	0	0
PROG_EV_01_01	监听与广播机制	监听与广播机制的概念以及应用	知道监听与广播机制的基本概念和使用	能够根据要求，实现监听与广播机制的典型应	能够根据问题需求，设计和实现监听与广播机制				

				用					
PROG_EV_01_02	事件	事件的概念以及应用	知道事件的基本概念和使用	能够根据要求，实现事件的定义和使用					
PROG_EV_01_03	事件源与事件监听器	事件源与事件监听器的概念以及应用	知道事件源与事件监听器的基本概念和使用	能够根据要求，实现事件源与事件监听器	能够根据问题需求，设计和实现事件源与事件监听器机制				
PROG_EV_02	事件驱动应用					3	2	1	0
PROG_EV_01_01	委托		知道委托的基本概念以及与事件的联系	能够根据要求，实现委托	能够根据问题需求，设计和实现委托				
PROG_EV_01_02	组件交互		知道组件交互的概念和应用	能够根据要求，实现基于事件驱动的组件交互	能够根据问题需求，设计和实现基于事件驱动的组件交互				

附　录　A
（资料性附录）
“C++语言程序设计”课程知识体系与教学要求

A.1 课程说明

以典型的“C++语言程序设计”课程知识体系与教学要求为例，展示如何使用标准规定的知识点及教学要求描述规范，说明一门课程的教学要求

本附录假设“C++语言程序设计”课程为高级语言程序设计的入门课程，完全针对零起点的学生，可作为其他信息类相关课程的基础课。目标是使学生通过本课程的学习，掌握面向对象程序设计的基本概念和方法、C++的基本语法和编程方法；学会使用集成开发环境；掌握程序调试方法；初步了解常用数据结构和非数值算法；初步了解C++标准模板库的使用方法。

A.2 教学要求

表A.1给出了课程知识单元与标准知识点的对应关系以及各知识点的教学要求。

表 A.1 “C++语言程序设计”课程知识体系与教学要求

<table>
<tr><th colspan="2" rowspan="3">课程知识单元</th><th colspan="6">对应标准知识点与教学要求</th></tr>
<tr><th rowspan="2">标识符</th><th rowspan="2">名称</th><th rowspan="2">要求</th><th>总学时</th><th>理论</th><th>实践</th></tr>
<tr><th colspan="3">教材推荐学时（标准学时）</th></tr>
<tr><td rowspan="4">前导知识</td><td rowspan="3">● 计算机系统
● 计算机语言和程序设计方法的发展
● 面向对象的基本概念
● 计算机中的信息与存储单位</td><td rowspan="3">无</td><td rowspan="3"></td><td rowspan="3">了解</td><td rowspan="4">3</td><td rowspan="4">2</td><td rowspan="4">1</td></tr>
<tr><td rowspan="3">● 程序的开发过程</td></tr>
<tr><td>PROG_FD_12</td><td>程序开发过程与工具环境</td><td>掌握</td></tr>
<tr><td>PROG_FD_12_01</td><td>程序的开发过程</td><td></td></tr>
<tr><td rowspan="9">C++简单程序设计</td><td rowspan="4">● C++语言概述</td><td>PROG_FD_01</td><td>基础语法</td><td>掌握</td><td rowspan="9">11</td><td rowspan="9">7</td><td rowspan="9">4</td></tr>
<tr><td>PROG_FD_01_01</td><td>顺序结构的程序</td><td></td></tr>
<tr><td>PROG_FD_01_02</td><td>字符集</td><td></td></tr>
<tr><td>PROG_FD_01_03</td><td>词法记号</td><td></td></tr>
<tr><td>● 数据的输入与输出</td><td>PROG_FD_01_04</td><td>标准输入/输出</td><td>掌握</td></tr>
<tr><td rowspan="3">● 基本数据类型和表达式</td><td>PROG_FD_02</td><td>数据类型</td><td>综合应用</td></tr>
<tr><td>PROG_FD_02_01</td><td>基本数据类型</td><td></td></tr>
<tr><td>PROG_FD_02_02</td><td>类型转换</td><td></td></tr>
</table>

		PROG_FD_03	数据的存储与运算	综合应用			
		PROG_FD_03_01	常量				
		PROG_FD_03_02	变量				
		PROG_FD_03_03	内置运算符				
		PROG_FD_03_04	表达式				
		PROG_FD_01_05	语句	综合应用			
	● 算法的基本控制结构	PROG_FD_04	流程控制语句	综合应用			
		PROG_FD_04_01	选择语句				
		PROG_FD_04_02	循环语句				
		PROG_FD_04_03	其他控制语句				
	● 自定义数据类型	无		掌握			
函数	● 函数的定义与使用 ● 内联函数	PROG_FD_05	功能模块	综合应用	5	3	2
		PROG_FD_05_01	功能模块的定义				
		PROG_FD_05_02	功能模块的调用				
	● 函数的参数传递 ● 带默认参数值的函数	PROG_FD_05_03	功能模块间的数据共享				
		PROG_FD_08	引用	综合应用			
	● C++系统函数	PROG_FD_11（一部分）	标准库	掌握			
	● 函数重载	PROG_00_01_01_03	函数/方法重载	综合应用			
类与对象	● 面向对象程序的基本特点	无		掌握	9	5	4
	● 类与对象	PROG_00_01	类与对象	综合应用			
		PROG_00_01_01	类的概念和定义				
		PROG_00_01_01_01	类的成员				
		PROG_00_01_01_02	成员的访问方式和权限				
		PROG_00_01_02	对象的概念和定义				
	● 构造函数 ● 析构函数	PROG_00_01_03	对象的初始化和删除				
	● 复制构造函数	PROG_00_01_04	对象的复制				
	● 对象作函数参数	PROG_00_01_05	对象在函数或方法间共享				
	● 类的组合	PROG_00_02_01	类的组合	综合应用			
		PROG_00_02_01_01	组合类的定义				
		PROG_00_02_01_02	组合类对象的初始化和删除				
		PROG_00_02_02	类之间的引用	综合应用			
数据的共享与保护	● 标识符的作用域与可见性	PROG_FD_05_03（一部分）	功能模块间的数据共享	综合应用	5	3	2

	● 对象的生存期						
	● 类的静态成员 ● 类的友元 ● 共享数据的保护	无		综合应用			
	● 多文件结构和编译预处理命令	无		了解			
数组、指针与字符串	● 数组的定义与初始化 ● 对象数组	PROG_FD_06	数组	综合应用	10	6	4
	数组作为函数的参数	PROG_FD_05_03（一部分）	功能模块间的数据共享				
	基于范围的 for 循环	PROG_FD_04_02（一部分）	循环语句				
	指针的定义和运算	PROG_FD_07	地址与指针（C/C++）	综合应用			
		PROG_FD_07_01	地址运算				
		PROG_FD_07_02	指针的定义				
		PROG_FD_07_03	指针运算				
		PROG_FD_07_04	指针的算术运算				
		PROG_FD_07_05	指针的关系运算				
		PROG_FD_07_06	指针的赋值运算				
	● 指针与函数 ● 对象指针 ● 动态内存分配 ● 智能指针 ● vector 对象 ● 对象深层复制与移动	无		综合应用			
	字符串	PROG_FD_09	字符串类型				
继承与派生	● 继承的基本概念和语法 ● 继承方式 ● 类型转换 ● 派生类的构造和析构 ● 派生类成员的标识与访问	PROG_00_02_03	类的继承	综合应用	5	3	2
		PROG_00_02_03_01	继承关系的定义				
		PROG_00_02_03_02	派生类对象的初始化和删除				
多态性	● 虚函数 ● 抽象类	PROG_00_02_04	多态性	综合应用	5	3	2
		PROG_00_02_04_01	多态性的概念和语法				

		PROG_OO_02_04_01	多态性的应用				
	● 运算符重载	无					
群体类和群体数据的组织	模板	PROG_GN_01	泛型程序设计	掌握	5	3	2
		PROG_GN_01_01	函数模板的概念和定义				
		PROG_GN_01_02	类模板的概念和定义				
		PROG_GN_01_03	模板的实例化				
		PROG_GN_01_04	模板的特殊实现				
	● 线性群体的概念 ● 数组 ● 链表 ● 栈 ● 队列 ● 排序 ● 查找	无		掌握			
泛型程序设计与 C++标准模板库		PROG_GN_02	C++中的泛型	了解	3	2	1(
	● 泛型程序设计及 STL 的结构	PROG_GN_02_01	STL 的概念和组成				
	● 迭代器	PROG_GN_02_02	迭代器的概念与定义				
	● 容器的基本功能与分类 ● 顺序容器 ● 关联容器	PROG_GN_02_03	容器的概念与定义				
	● 函数对象	PROG_GN_02_04	函数对象的概念与定义				
	● 算法	PROG_GN_02_05	STL算法				
流类库与输入/输出		PROG_FD_10	输入/输出与文件	掌握	4	2	2
	● I/O 流的概念及流类库结构	PROG_FD_10_01	输入/输出的基本概念和语法				
	● 输出流 ● 输入流 ● 输入/输出流	PROG_FD_10_02	文件处理				
异常处理	● 异常处理的思想与程序实现 ● 异常处理中的构造与析构 ● 标准程序库异常处理	PROG_OO_03	异常处理	了解	2	1	1
		PROG_OO_03_01	异常处理的基本思想				
		PROG_OO_03_02	异常处理的语法				

附　录　B
（资料性附录）
吉林大学本科 C#语言程序设计课程教学大纲对标准覆盖情况案例

B.1 课程说明

课程理论教学32学时

B.2 教学要求

表B.1给出了课程知识单元与标准知识点的对应关系以及各知识点的教学要求。

表 B.1 “C#语言程序设计”课程知识体系与教学要求

知识单元1（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
.NET 应用体系结构（1 学时）	如何选择你的技术		了解	0.25			0.25
	.NET 历史		了解	0.25			0.25
	*.NET 2015		掌握	0.25	0.25		
	Hello, world	PROG_FD_01_01	了解	0.25	0.25		
	*使用.NET 4.6 编译		掌握	0.25	0.25		
	使用.NET Core CLI 编译		掌握	0.25	0.25		
	应用种类和技术		了解	0.25			0.25
	开发工具	PROG_FD_12	了解	0.25			0.25
单元要点	掌握.NET 体系和重要组成部件，以及 CLI 的地位和作用。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。						

知识单元2（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
核心 C#（3 学时）	C#基础	PROG_FD_03	掌握	0.25	0.25		
	使用 Visual Studio 开发	PROG_FD_12	掌握	1	0.5	0.5	
	使用变量	PROG_FD_03_01	掌握	0.25			0.25
	使用预定义类型		掌握	0.25	0.25		
	控制程序流	PROG_FD_04	综合应用	0.5		0.25	0.25
	使用枚举		掌握	0.25	0.25		
	使用命名空间进行组织		掌握	0.25	0.25		
	*理解 Main 方法	PROG_FD_05	掌握	0.5	0.5		
	使用注释		掌握	0.25	0.25		

	*理解 C#预处理器指令		掌握	0.5	0.25	0.25	
	*C#编程规范		掌握	1	0.5	0.5	
单元要点	解 C#语言的核心概念和基本属性，理解 Main 方法以及基本的预处理指令，掌握 C#编程规则规范。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。						

知识单元3（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
对象和类型（3 学时）	*构建和使用类	PROG_OO_01_01 PROG_OO_01_02	综合应用	1	0.5	0.5	
	类和结构体概述		理解	0.25	0.25		
	*类	PROG_OO_02	综合应用	1	0.5	0.5	
	匿名类型		掌握	0.25	0.25		
	结构体		综合应用	0.25			0.25
	*以值和引用类型传递参数	PROG_FD_09	掌握	1	0.5	0.5	
	部分类		掌握	0.25	0.25		
	扩展方法		掌握	0.25	0.25		
	*Object 类		掌握	1	0.5	0.5	
单元要点	掌握类的基本概念，综合应用值和引用类型的使用方法，掌握 object 类的作用。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。						

知识单元4（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
继承（3 学时）	*继承概述	PROG_OO_02_03	了解	0.5	0.5		
	*继承的类型	PROG_OO_02_03	掌握	1	0.5	0.5	
	*继承的实现	PROG_OO_02_03 PROG_OO_02_04	综合应用	2	1	1	
	修饰符		掌握	0.25	0.25		
	接口继承		掌握	0.25	0.25		
	*is 和 as 运算符		掌握	1	0.5	0.5	
单元要点	掌握继承的基本概念，综合应用继承特性的使用方法，掌握 is 和 as 运算符的作用。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。						

知识单元	知识要求	对应标准点	能力要求	学时要求

5（章节）				总	理论	实践	其他
泛型（6学时）	泛型概述		了解	0.5	0.5		
	*泛型类	PROG_GN_01_02	综合应用	3	2	1	
	*泛型特性	PROG_GN_01_03	综合应用	2	1	1	
	泛型接口		综合应用	2	1	1	
	泛型结构体		综合应用	0.5	0.5		
	*泛型方法	PROG_GN_01_01	综合应用	2	1	1	
单元要点	了解泛型基本概念，能综合应用泛型类及其特性，以及泛型接口、结构体和方法。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。						

知识单元6（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
数组和元组（2学时）	同类和异类的多个对象		了解	0.25			0.25
	简单数组	PROG_FD_06	掌握	0.25			0.25
	*多维数组	PROG_FD_06	掌握	0.5	0.25	0.25	
	锯齿数组		了解	0.25			0.25
	*Array 类		综合应用	2	1	1	
	*数组作为参数	PROG_FD_06	掌握	0.5	0.25	0.25	
	枚举	PROG_FD_09	掌握	0.25			0.25
	元组		掌握	0.25			0.25
	*结构比较		掌握	0.5	0.25	0.25	
单元要点	掌握数组和元组的基本特性和基本方法，综合运用内置数据结构。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。						

知识单元7（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
运算符和类型转换（2学时）	运算符和转换概述		了解	0.25			0.25
	*运算符	PROG_FD_03_03	掌握	0.5	0.5		
	*类型的安全性	PROG_FD_02_02	掌握	0.5	0.25	0.25	
	比较对象的相等性		掌握	0.5	0.25	0.25	
	*运算符重载		综合应用	1	0.5	0.5	
	数组下标运算符实现		掌握	0.25		0.25	
	*用户自定义类型转换		综合应用	1	0.5	0.5	
单元要点	掌握运算符和类型转换的基本特性和方法，综合应用自定义转换方法。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。						

知识单元8（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
委托与事件（6 学时）	委托概述	PROG_EV_02_01	了解	0.25			0.25
	委托类	PROG_EV_02_01	掌握	0.5	0.5		
	委托的实现	PROG_EV_02_01	综合应用	2	1	1	
	系统预定义委托	PROG_EV_02_01	综合应用	0.25	0.5		
	委托的应用	PROG_EV_02_01	综合应用	2	1	1	
	Lambda 表达式		综合应用	1	0.5	0.5	
	事件概述	PROG_EV_01_02	了解	0.5	0.5		
	监听与广播机制	PROG_EV_01_03	掌握	1	1		
	委托在事件处理中的应用	PROG_EV_01_05	综合应用	1.5	1	0.5	
单元要点	掌握委托的基本特性和基本方法，综合应用 Lambda 表达式和事件机制。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。						

知识单元9（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
字符串（1学时）	System.String 类	PROG_FD_10_02	掌握	0.25	0.25		
	*字符串格式		掌握	0.25	0.25		
	*正则表达式		综合应用	1	0.5	0.5	
单元要点	掌握正则表达式的基本特性和基本方法，综合应用相关特性。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。						

知识单元10(章节)	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
聚集（3学时）	聚集概述		了解	0.25			0.25
	*聚集接口和类型		掌握	0.25	0.25		
	*列表		综合应用	1	0.5	0.5	
	队列		掌握	0.25			0.25
	*栈		掌握	0.25	0.25		
	*链接列表		掌握	0.25	0.25		
	有序列表		掌握	0.25	0.25		
	*字典		综合应用	1	0.5	0.5	
	集合		掌握	0.25	0.25		
	*性能		掌握	0.25	0.25		
	*使用 bit		掌握	1	0.25	0.25	0.5
	可见聚集		了解	0.25			0.25
	不可修改聚集		了解	0.25			0.25
	并发聚集		了解	0.25	0.25		

单元要点	掌握多种样式的数据聚集的基本特性和基本方法，灵活使用相关特性。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。

知识单元11(章节)	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
错误与异常（2 学时）	异常概述	PROG_OO_03_01	了解	0.25			0.25
	*异常类		掌握	0.25	0.25		
	*捕获异常	PROG_OO_03_02	掌握	1	0.5	0.5	
	*用户定义的异常类		综合应用	2	1	1	
	调用者信息		掌握	0.25	0.25		
单元要点	掌握异常的基本特性和基本方法，综合应用掌握捕获异常的方法，以及相关特性。*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。						

附　录　C
（资料性附录）
北京理工大学本科 C 语言程序设计课程教学大纲规范案例

C.1 课程说明

本课程是面向计算机“零起点”学生开设的高级语言程序设计入门课程，是本科理工科信息类公共基础必修课。

本课程目的：以 C 语言为背景掌握程序设计的基础知识，掌握基本的程序设计过程和技能，基本编程思想和方法，培养学生使用计算机处理实际问题的意识和能力，使学生具备初步的程序设计能力，熟练掌握一种集成开发环境的使用方法，具备调试 C 语言程序的一般技能。为培养理论与实践相结合的信息技术人才打下良好的基础。

C.2 C语言程序设计课程教学内容

课程总学时 64 学时（其中教学 32 学时，上机实践 32 学时）。表中*为重要知识点。“其他”学时以学生课后自学形式要求，不计入教学学时数量。

表 C.1“C 语言程序设计”课程知识体系与教学要求

知识单元 1（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
程序设计语言入门（3 学时）	C 语言诞生的背景		了解		2	1	2
	高级语言程序的一般执行过程		了解				
	*集成开发环境 Dev C 演示		掌握				
	C 语言程序的基本结构和组成：函数结构、注释等		了解				
	程序设计语言的基本组成：基本字符集、标识符、关键字		了解				
	*数据类型：int 型的常量和变量、scanf、printf、格式控制符%d		掌握				
	*顺序程序结构		掌握				
	运算符：赋值运算=，算术运算(、)、+、-、、/、%，取地址 &		掌握				
单元要点	理解高级语言程序的开发过程，掌握 IDE 的基本使用方法，掌握对 scanf 和 printf 函数输入/输出 int 型变量和表达式的方法，能够采用顺序结构编写不超过 10 行的 C 语言程序。						

知识单元 2	知识要求	对应标准点	能力	学时要求

（章节）			要求	总	理论	实践	其他
分支结构 if（3 学时）	＊if-else 语句的两种基本语法和执行流程		掌握	3	1.5	1.5	3
	＊ 运算符：关系运算 >、>=、<、<=、==、!=，逻辑运算 &&、||、!		掌握				
	＊ 嵌套的 if-else 语句的匹配规则，程序实例		掌握				
	将数学描述转换为逻辑表达式		了解				
	＊ if 程序实例		综合应用				
	运算符：条件运算 ？：		掌握				
单元要点	掌握 if 语法和执行流程，掌握逻辑运算和逻辑表达式，掌握嵌套的 if 语句中 if-else 的配对规则，能够根据实际问题正确书写逻辑表达式，能够规范使用嵌套的 if 语句编写不超过 20 行的 C 语言程序。						

知识单元 3（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
分支结构 switch（3 学时）	＊ switch 语句的基本语法和程序执行流程		掌握	3	1	2	3
	将现实问题转换为逻辑表达式		了解				
	讨论：break 在 switch 语句中的作用，switch 与 if 的比较		掌握				
	＊ switch 程序实例		综合应用				
	数据类型：double 型的常量和变量、格式控制符%f 和%lf		了解				
单元要点	掌握 switch 语法和执行流程，掌握 switch 语句与 break 语句的关系，掌握 switch 与 if 的区别，能够规范使用 switch 语句编写不超过 20 行的 C 语言程序。						

知识单元 4（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
循环结构 while（4 学	＊while 语句的基本语法和程序执行流程		掌握	3	2	2	4

时）	* while 程序实例		综合应用				
	运算符：自增自减运算 ++、--		了解				
	* 数据类型：char 型的常量和变量、格式控制符%c、函数 getchar 和 putchar		掌握				
单元要点	掌握 while 语法和执行流程，能够针对应用问题规范使用 while 语句编写不超过 20 行的一重循环程序。						

知识单元 5（章节）	**知识要求**	**对应标准点**	**能力要求**	**学时要求**			
				总	**理论**	**实践**	**其他**
循环结构 for 和 do-while（3 学时）	* for 语句的基本语法和程序执行流程，for 语句与 while 关系		掌握	3	1.5	1.5	3
	for 程序实例		掌握				
	* do-while 语句的基本语法和程序执行流程，do-while 语句与 while 关系		掌握				
	do-while 程序实例		掌握				
	运算符：逗号运算 ,，赋值运算 =、op=		了解				
单元要点	掌握 for 和 do-while 语法和执行流程，掌握三种循环语句之间的逻辑关系，能够针对应用问题规范使用 for 和 do-while 语句编写不超过 20 行的一重循环程序。						

知识单元 6（章节）	**知识要求**	**对应标准点**	**能力要求**	**学时要求**			
				总	**理论**	**实践**	**其他**
循环嵌套（4 学时）	* 多重循环嵌套的执行流程和基本规则		掌握	4	2	2	4
	多重循环程序实例		掌握				
	* 转移语句：break 语句的作用与执行流程		掌握				
	* 转移语句：continue 语句作用与执行流程		掌握				
	转移语句：goto 语句作用与执行流程		了解				

单元要点	掌握 break 和 continue 语法和执行流程，了解 goto 语言的语法和使用注意，掌握在循环中消除 break 和 continue 的方法，能够针对应用问题使用循环语句编写不超过 30 行的多重循环程序。

知识单元 7（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
简单算法的设计（4 学时）	编写程序的一般步骤		掌握	4	2	2	4
	*数值型问题实例：穷举法		综合应用				
	*数值型问题实例：递推法						
	*非数值问题实例：逐步求精						
单元要点	掌握简单应用问题编写程序的一般步骤，掌握典型的数值型问题的求解过程和典型算法（穷举法和递推法）的程序实现过程，掌握逐步求精法的思想，能够解决典型的非数值型问题（输出由字符组成的特定图形），能够针对应用问题进行简单算法设计、并编写不超过 30 行的程序解决应用问题。						

知识单元 8（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
数组（4 学时）	* 数组基本概念		掌握	4	2	2	4
	* 一维数组的说明、数组大小、元素的引用、下标和下标运算符[ ]		掌握				
	一维数组的输入、输出		掌握				
	一维数组应用实例：冒泡排序等		掌握				
	* 二维数组的说明、元素引用		掌握				
	二维数组应用实例		掌握				
单元要点	掌握数组的说明、数组大小、元素引用、下标和下标运算符等基本概念，掌握一维和二维（数值型）数组的使用方法，能够采用数组解决简单应用问题。掌握常见的排序方法（冒泡法）并编写程序。						

知识单元 9（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
字符数组（4 学时）	* 字符串基本概念：字符串、串结束标记\0、串长、字符数组		掌握	4	2	2	4

	一维字符数组的说明、数组元素的引用		掌握				
	字符串的整体输入输出：scanf 和 printf 的格式控制符%s、gets 和 puts		掌握				
	* 字符串基本操作：求串长、串复制、串连接、串比较、求子串等		掌握				
单元要点	掌握字符串、串结束标记和串长的基本概念，掌握常见字符串操作（求串长、串复制、串连接、串比较、求子串等）的基本原理和算法，能够编程实现。						

知识单元 10（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
函数（4 学时）	C 程序的一般结构		掌握	4	2	2	4
	* 函数的定义、说明、调用和返回		掌握				
	* 函数间参数传递规则：值传递、返回值		掌握				
	应用实例：swap 函数		掌握				
	数据类型：void		掌握				
	* 变量存储类型：自动变量、寄存器变量、外部变量和静态变量		掌握				
单元要点	掌握函数的定义、说明、调用和返回，掌握函数间参数传递的规则，掌握变量存储类型的特点，能够编写一个简单的不超过 10 行的自定义函数并正确调用。						

知识单元 11（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
函数递归（4 学时）	* 递归的基本概念		掌握	4	2	2	4
	* 递归程序的执行过程		掌握				
	* 求阶乘的执行过程		掌握				
	* 反向输出整数的执行过程		掌握				
	汉诺塔问题的执行过程		了解				
	递归程序的理论基础讨论		掌握				
单元要点	掌握递归的基本概念，能够识别递归程序并分析不超过 10 行的直接递归程序的						

	执行过程。

知识单元 12（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
递归编程方法（4 学时）	*数值型问题的递归编程方法		掌握	4	2	2	4
	*非数值型问题的递归编程方法		掌握				
单元要点	掌握直接递归程序的编写方法，能够针对具有递归性质的问题建立递归模型（过程），并正确编写不超过 20 行的直接递归函数。						

知识单元 13（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
指针（4 学时）	* 指针基本概念：变量的基本特征（变量名，数据类型，值和地址），指针变量说明，指针赋值，通过指针访问变量		掌握	4	2	2	4
	* 与指针相关的运算符：&、*、++、--		掌握				
	* 指针与对象的关系及典型应用		掌握				
	* 指针与变量：指针变量、指针赋值、直接访问、间接访问		掌握				
	* 指针与函数：指针参数（传递地址）、swap 函数、函数返回指针、函数指针等		了解				
	* 指针与数组：指针数组、数组指针、main 函数的参数、字符串处理、动态存储空间分配与释放 malloc 和 free 等		掌握				
	指针与指针：指针型指针等						
单元要点	掌握指针的基本概念与相关运算，在掌握基本理论的基础上，能够综合应用相关知识编写长度不超过 30 行的基于指针的字符串处理函数。						

知识单元 14（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
综合算法（4 学时）	* 排序算法：选择排序、插入排序、字符串排序		掌握	4	2	2	4

	贪心法		掌握				
单元要点	能够针对应用问题进行分析建模，并综合运用数组、函数、指针等知识，进行合理的功能划分并用多个函数完成，实现长度不超过 80 行的程序。						

知识单元 15（章节）	**知识要求**	**对应标准点**	**能力要求**	**学时要求**			
				总	**理论**	**实践**	**其他**
自定义数据类型 1（4 学时）	* 结构基本概念：定义、说明和使用		掌握	4	2	2	4
	结构应用实例		掌握				
	运算符：.、->、sizeof		掌握				
	自定义类型：typedef		掌握				
	* 结构与其他对象的关系：指针成员、结构数组、结构指针		了解				
	结构与指针：指针作为结构成员，指向结构的指针（结构指针） 结构与数组：结构数组 结构与函数：结构作为函数的形参，函数返回结构、返回指向结构指针		掌握				
单元要点	掌握结构的基本概念和相关基本理论，能够综合应用结构、指针、数组和函数的能够知识针对应用问题编写不超过 80 行的应用程序。						

知识单元 16（章节）	**知识要求**	**对应标准点**	**能力要求**	**学时要求**			
				总	**理论**	**实践**	**其他**
自定义数据类型 2（0 学时）	联合基本概念：定义、说明和使用		了解	0	0	0	2
	枚举基本概念：定义、说明和使用		了解				
	简单应用实例		掌握				
单元要点	掌握联合、枚举的基本概念和相关基本理论。						

知识单元 17	**知识要求**	**对应标准点**	**能力**	**学时要求**

（章节）			要求	总	理论	实践	其他
链表 （4 学时）	* 链表的基本概念		掌握	4	2	2	4
	* 单链表基本操作的实现：创建、插入、查找、删除		掌握				
	典型应用实例		掌握				
单元要点	掌握链表的基本概念，能够编程实现单链表的基本操作，并能够针对应用问题运用链表知识编写不超过 100 行的程序求解问题。						

知识单元 18 （章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
文件 （4 学时）	* 文件基本概念：文件分类、一般读写过程		掌握	4	2	2	4
	文件的打开与关闭：FILE、fopen、fclose、标准设备文件		掌握				
	文件的顺序读写：fgetc/fputc、fgets/fputs、fprintf/fscanf、fread/fwrite		掌握				
	文件的随机读写：fseek、ftell、rewind、feof、ferror、clearerr		掌握				
	文件操作综合实例		综合应用				
单元要点	掌握文件的基本概念，能够正确应用文件基本操作函数完成对文件的顺序/随机访问，能够根据应用问题设计文件的数据格式，并完成不超过 150 行的文件处理程序。						

知识单元 19 （章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
位运算 （0 学时）	位运算符：|、&、^、~、<<、>>		了解	0	0	0	2
	应用实例		了解				
单元要点	掌握位运行的基本概念和相关运算符。						

知识单元 20（章节）	知识要求	对应标准点	能力要求	学时要求			
				总	理论	实践	其他
预处理（0 学时）	预处理命令：#include、#define		了解	0	0	0	2
	应用实例		了解				
单元要点	掌握 C 语言预处理的基本命令，了解编译器的预处理过程，在程序中能够正确使用基本预处理命令。						

附　录　D
（资料性附录）
“C 语言程序设计”（线上线下混合）课程教学要求（案例）

D.1　课程说明

本案例以四川师范大学《高级语言程序设计》课程线上线下混合式教学为例，展示如何使用标准规定的知识点及教学要求描述规范，说明一门课程的教学要求。

本校《高级语言程序设计》课程为计算机类大一学生第一门专业核心课程，课程以C语言为实践语言，完全针对零起点的学生，可作为其他信息类相关课程的基础课。目标是使学生通过本课程的学习，掌握C语言程序设计语法知识，学会阅读程序，能够编写程序并能熟练使用C语言开发环境调试程序；具备设计实验项目的能力，能够分析实验结果，得到合理有效结论，形成报告；具备数学抽象归纳、逻辑思考和推理能力，能够将数学模型用C语言程序描述并进行问题求解；具备运用程序设计方法解决系统设计和集成等工程问题的能力。

D.2　教学要求

本校《高级语言程序设计》课程学时共72学时，分为：理论48学时，实验24学时。其中，理论和实验均采用线上线下混合式教学，纳入教学计划线上学时为16学时，占总学时22%，此部分要求学生必须线上完成，其余线上自主学习学生可根据自身对知识点掌握情况进行有选择学习，但最终线上学习成绩和数据将计入到期末考核总成绩中。

表A.1给出了课程理论教学章节与标准知识点的对应关系以及各知识点的教学要求；

表A.2给出了课程实验教学内容与标准知识点的对应关系以及各知识点的教学要求；

表A.3给出了课程考核与标准知识点的对应关系。

具体各表如下所示：

D.2.1 课程理论教学章节与标准知识对应表

表 D.1 《高级语言程序设计》课程理论教学章节与标准知识对应表

章标题	主要内容	教学要求	学时	标准知识点标识符及学时	教学方法
第 1 章 为什么要学 C 语言	1. 引言 2. 游戏、黑客和 C 语言 3. C 语言，不老传说 4. C 语言教给我们的事 5. 什么是编程	1. 理解学习 C 语言的重要性 2. 理解程序设计思想	1(线下)	PROG_FD_01_01 0 学时	课堂讲授 线上学习
第 2 章 C 数据类型	1. 常量与变量 2. 简单的屏幕输出 3. 数据类型 4. 如何计算变量或数据类型所占内存空间的大小	1. 掌握 C 语言的基本数据类型 2. 理解类型的意义及不同类型的区别	2(线下)	PROG_FD_02 2 学时	课堂讲授 线上学习

章标题	主要内容	教学要求	学时	标准知识点标识符及学时	教学方法
	5. 变量的赋值和赋值运算符				
第 3 章 简单的算术运算和表达式	1. C 运算符和表达式 2. 算术运算符和表达式 3. 复合的赋值运算符 4. 增 1 和减 1 运算符 5. 宏常量与宏替换 6. const 常量 7. 自动类型转换与强制类型转换运算符. 8. 常用的标准数学函数	1. 理解 C 语言的运算符及构成表达式的规则 2. 掌握 C 语言包含的运算符及含意 3. 理解运算符的优先级及结合性，类型转换的规则，增 1 和减 1 运算符	3(线下)	PROG_FD_03 2 学时	讲练结合 线上学习
第 4 章 键盘输入和屏幕输出	1. 单个字符的输入 / 输出 2. 数据的格式化屏幕输出 3. 数据的格式化键盘输入	1. 理解 C 语言的输入输出功能的实现 2. 掌握格式输出和输入的函数及使用 3. 能解决输入时可能会遇到的输入错误问题	1(线下)	PROG_FD_01_04 1 学时	讲练结合 线上学习
第 5 章 选择控制结构	1. 生活中与计算机中的问题求解方法 2. 算法的概念及描述方法 3. 关系运算符、运算表达式 4. 用于单分支控制的条件语句★ 5. 用于双分支控制的条件语句★ 6. 条件运算符、条件表达式★ 7. 用于多分支控制的条件语句★ 8. 用于多路选择的 switch 语句 9. 逻辑运算符、逻辑表达式	1. 理解选择结构对应程序的意义，掌握实现选择结构的语法 2. 掌握用于单分支控制的条件语句，用于双分支控制的条件语句 3. 掌握用于多路选择的 switch 语句	3(线下)	PROG_FD_05_01 2 学时	课堂讲授 讲练结合 线上学习
第 6 章 循环控制结构	1. 循环控制结构与循环语句★ 2. 计数控制的循环 3. 嵌套循环★	1. 理解循环结构的意义及实现的语法 2. 掌握三种循环语句的使用、区别和联系	3(线下)	PROG_FD_05_12 3 学时	课堂讲授 讲练结合 线上学习

章标题	主要内容	教学要求	学时	标准知识点标识符及学时	教学方法
	4. 条件控制的循环 5. 流程的转移控制	3. 掌握 break 语句、continue 语句、穷举法			
第 7 章 函数	1. 分而治之与信息隐藏 2. 函数的定义★ 3. 向函数传递值和从函数返回值★ 4. 函数的递归调用和递归函数▲ 5. 变量的作用域和存储类型▲ 6. 模块化程序设计	1. 理解函数的意义和语法，学习模块化程序设计的思想 2. 掌握函数的定义和调用，函数的参数传递，递归函数等 3. 能实现递归函数 4. 理解变量作用域，结构化程序设计思想	4 (线下) + 2(线上)	PROG_FD_06 4 学时	课堂讲授 翻转课堂 线上学习
第 8 章 数组	1. 一维数组的定义和初始化 2. 二维数组的定义和初始化 3. 向函数传递一维数组★ 4. 排序和查找▲ 5. 向函数传递二维数组★	1. 理解数组的意义和存储方式，数组的定义和使用的语法，用数组辅助解决的问题 2. 掌握数组的定义和使用，一维和二维数组 3. 掌握数组元素的初始化，排序和查找，向函数传递二维数组	4 (线下) + 2(线上)	PROG_FD_07_01 PROG_FD_07_02 4 学时	课堂讲授 翻转课堂 线上学习
第 9 章 指针	1. 变量的内存地址 2. 指针变量的定义和初始化 3. 间接寻址运算符★ 4. 按值调用与模拟按引用调用▲ 5. 用指针变量作函数参数的程序实例▲ 6. 函数指针及其应用	1. 理解指针数据类型，指针的语法及指针的应用 2. 理解内存地址、指针变量的定义和初始化等 3. 掌握用指针变量作函数参数，函数指针	4(线下)	PROG_FD_08 6 学时	课堂讲授 讲练结合 线上学习
第 10 章 字符串	1. 字符串常量 2. 字符串的存储 3. 字符指针★ 4. 字符串的访问和输入 / 输出 5. 字符串处理函数★ 6. 向函数传递字符串▲ 7. 从函数返回字符串指针▲	1. 理解字符串的存储，常用操作及函数，输入输出及字符串和函数的结合 2. 能正确使用字符串的存储，字符串处理函数 3. 理解向函数传递字符串、从函数返回字符串指针	4(线下)	PROG_FD_10 1 学时	案例讨论 讲练结合 线上学习
第 11 章 指针和数	1. 指针和一维数组间的关系★	1. 理解指针和数组的关系，以及指针和数组的互用性	4(线下) +	PROG_FD_08	课堂讲授 翻转课堂

章标题	主要内容	教学要求	学时	标准知识点 标识符及学时	教学方法
组	2. 指针和二维数组间的关系★ 3. 指针数组及其应用▲ 4. 动态数组▲	2. 理解指针和数组的关系，指针数组用于表示多个字符串 3. 掌握动态数组，动态内存分配函数，长度可变的一维动态数组	2(线上)	6 学时	线上学习
第 12 章 结构体和 共用体	1. 从基本数据类型到抽象数据类型 2. 结构体的定义 3. 结构体数组的定义和初始化 4. 结构体指针的定义和初始化 5. 向函数传递结构体★ 6. 共用体 7. 枚举数据类型 8. 动态数据结构——单向链表▲	1. 理解结构体和共用体的含义及语法，学习结构体在解决问题中的应用 2. 掌握结构体的语法和初始化，结构体做参数 3. 能正确使用结构体指针、共用体、枚举、单向链表解决实际问题	6(线下)	无	课堂讲授 讲练结合 线上学习
第 13 章 文件操作	1. 二进制文件和文本文件 2. 文件的打开和关闭 3. 按字符读写文件★ 4. 按格式读写文件★ 5. 按数据块读写文件★ 6. 文件的随机读写 7. 标准输入 / 输出重定向	1. 理解文件操作的意义，文件操作的方法，文件的类型 2. 掌握文件的打开和关闭、按字符读写文件、按格式读写文件、按数据块读写文件 3. 掌握按数据块读写文件、文件的随机读写	3(线下)	无	课堂讲授 分组讨论 线上学习

注：★表示重点内容，▲表示难点内容

D.2.2 课程实验教学内容与标准知识对应表

表 D.2 《高级语言程序设计》课程实验教学内容与标准知识对应表

序号	实验名称	实验内容	学时	评价依据	标准知识点 标识符及学时

序号	实验名称	实验内容	学时	评价依据	标准知识点 标识符及学时
1	数据类型和表达式	1. 计算由键盘输入的任何两个整型数据，求他们的平均值并输出到屏幕 2. 输入存款金额，存期和年利率，计算存款到期时的利息，输出时保留 2 位小数	2(线上)	实验工作记录、实验报告	PROG_FD_02 PROG_FD_03 4 学时
2	程序的结构	1. 了解和掌握分支语句的使用，包括 if 语句的各种形式以及 switch 语句 2. 使用循环语句完成累乘、图像输出的程序编写 3. 掌握较复杂结构程序的编写 4. 掌握程序调试的方法	4(线下) + 2(线上)	实验工作记录、实验报告	PROG_FD_05 6 学时
3	过程考核一	C 语言基础知识上机考试	不占学时 考试 2 个小时	单独测试	PROG_FD_01 PROG_FD_02 PROG_FD_03 PROG_FD_04 PROG_FD_05
4	函数	1. 理解函数的编程思想，编写一个包括 3~4 个函数的程序 2. 掌握函数中参数传递的两种方式和函数的相互调用	2(线下) + 2(线上)	实验工作记录、实验报告	PROG_FD_06 4 学时
5	数组	1. 掌握一维和二维数组的使用 2. 能运用一维和二维数组解决实践问题	2(线下) + 2(线上)	实验工作记录、实验报告	PROG_FD_07 4 学时
6	过程考核二	C 语言核心知识	不占学时 考试 2 个小时	单独测试	PROG_FD_06 PROG_FD_07
7	指针	1. 用指针作为函数参数完成字符串的传递 2. 掌握函数中参数传递的两种方式 3. 掌握指针与数组的关系，以及使用指针操作数组	2(线下) + 2(线上)	实验工作记录、实验报告	PROG_FD_08 PROG_FD_10 7 学时
9	过程考核三	C 语言综合知识及应用	不占学时 考试 2 个小时	单独测试	PROG_FD_08 PROG_FD_10
8	结构体和文件	1. 掌握结构体的定义、数据输入方法 2. 掌握结构体成员变量的两种访问方法 3. 掌握编写程序完成单链表的建立和查询方法 4. 掌握文件的打开和关闭 5. 精通文件的读写操作 6. 了解文件的定位操作及文件的检测函数 7. 掌握文件的应用	4(线下)	实验工作记录、实验报告	无

D.2.3 课程考核与标准知识对应表

表 D.3 《高级语言程序设计》课程考核与标准知识对应表

考核依据	分值	考核内容/评价细则	标准知识点标识符	形成文档资料
考勤	5 分	基准分 100 分，每缺一次扣 20 分，扣到 0 分为止	无	教学记录册
线上作业	20 分	1. 内容：主要考核学生对每章节知识点的自主学习能力、复习、理解和掌握程度 2. 评价：每周 SPOC 平台上作业分为单元测验和单元作业，其中，单元测验成绩占总分的 60%，单元作业成绩占总分的 40%	PROG_FD_01 PROG_FD_02 PROG_FD_03 PROG_FD_04 PROG_FD_05 PROG_FD_06 PROG_FD_07 PROG_FD_08 PROG_FD_10 链表、结构体和文件知识点（无）	作业成绩册
过程考核	15 分	1. 内容：主要考核学生对每章节知识点的复习、理解和掌握程度 2. 评价：每次过程考核按 100 分制打分，最后折算每次过程考核满分 5 分，三次共计 15 分	PROG_FD_01 PROG_FD_02 PROG_FD_03 PROG_FD_04 PROG_FD_05 PROG_FD_06 PROG_FD_07 PROG_FD_08 PROG_FD_10	过程考核试卷 源代码或作品 过程考核成绩册
实验	10 分	1. 内容：主要考核学生对每章节知识点的复习、理解和掌握程度 2. 评价：每次实验报告格式正确，针对实验内容正确完成，根据等级优、良、中、及格和不及格打成绩，最终折算分值为：实验报告 1、2 各 1 分，实验报告 3、4、5、6 各 2 分，共 10 分	PROG_FD_01 PROG_FD_02 PROG_FD_03 PROG_FD_04 PROG_FD_05 PROG_FD_06 PROG_FD_07 PROG_FD_08 PROG_FD_10 结构体和文件知识点（无）	实验报告 源代码或作品 实验成绩册

考核依据	分值	考核内容/评价细则	标准知识点标识符	形成文档资料
期末考试	50 分	1. 内容：主要考核学生对每章节知识点的复习、理解和掌握程度 2. 评价：期末成绩按 100 分制打分，最终将期末笔试成绩折算为满分 50 分对应成绩	PROG_FD_01 PROG_FD_02 PROG_FD_03 PROG_FD_04 PROG_FD_05 PROG_FD_06 PROG_FD_07 PROG_FD_08 PROG_FD_10 链表、结构体和文件知识点（无）	期末试卷(AB) 试卷审阅表 期末考试成绩册 试卷分析报告

附　录　E
（资料性附录）
天津师范大学本科“JAVA 语言程序设计”课程规范案例

E.1 课程说明

本课程是针对计算机类学生开设的第二门程序设计类课程，学生已经完成“C语言程序设计”或者“C++程序设计”等一门及以上相关程序设计类课程，开设学期一般为大学一年级下学期及后续学期。学生已经具有一定的计算思维、问题求解和结构化程序设计能力，开始系统深入学习面向对象程序设计。课程的教学以课内讲授与课外练习、上机实验相结合的方式进行。教学内容主要包括：掌握Java语法及面向对象的基础知识；较熟练地编写Java应用程序；理解集合框架、事件驱动、异常处理、多线程；了解Java的常用标准类库、Java Swing、Socket等。要求学生熟练掌握Java程序设计的基础知识、基本概念；掌握面向对象程序设计的思想和编程技巧；熟练掌握分析程序和运用面向对象的思路和方法编制程序的能力。

E.2 教学要求

表E.1给出了课程知识单元与标准知识点的对应关系以及各知识点的教学要求。

表 E.1 《JAVA语言程序设计》与课程标准知识对应表

知识单元			对应标准知识点与教学要求					
名称	具体要求	主要内容	标识符	名称	能力要求	总学时	理论	实践
1.概述	了解 Java 特性；掌握 JDK 运行环境;学会帮助文件的使用;学会编写并运行最简单的 Java 程序。	1.1Java 的历史	无		了解	2	2	0
		1.2Java 的特性：跨平台、JVM、垃圾回收、解释型等。	无		了解			
		1.3Java 的应用领域	无		了解			
		1.4JDK 的安装与配置	PROG_FD_12_02	开发环境/工具	掌握			
		1.5 一个简单的 Java 程序	PROG_FD_01_01	顺序结构的程序	了解			
			PROG_FD_01_04	标准输入/输出	了解			
			PROG_FD_12_01	程序的开发过程	掌握			

		1.6 Java 文档及生成	无		了解			
2. 面向对象程序设计方法学	了解从结构化程序设计到面向对象程序设计进化过程；掌握面向对象程序设计中基本概念、基本思路；能够利用面向对象程序设计方法学来分析现实问题。	2.1 面向过程程序设计方法：功能分解、逐步求精，模块化，面向过程程序设计方法的产生、分析问题思路、解决问题过程、优缺点等。	无		了解	2	2	0
		2.2 面向对象程序设计方法：封装性、继承性、多态性，面向对象程序设计方法的产生、分析问题思路、解决问题过程、优缺点等。	无		掌握			
		2.3 面向对象程序设计的基本概念：类、对象、消息、类变量和实例变量、类方法和实例方法、继承等。	PROG_00_01	类与对象	了解			
			PROG_00_01_01	类的概念和定义	了解			
			PROG_00_01_01_01	类的成员	了解			
			PROG_00_01_02	对象的概念和定义	了解			
			PROG_00_02_03_01	继承关系的定义	了解			
3 基本语法	掌握 Java 程序的基本组成；掌握 Java 程序的简单输入输出语句；掌握数据类型及常量、变量等的用法；掌握各类运算符及其运算优先级；理解结构	3.1 Java 基本语法格式：标识符、关键字、常量、注释和程序书写规范。	PROG_FD_01_02	字符集	掌握	6	4	2
			PROG_FD_01_03	词法记号	掌握			
			PROG_FD_03_01	常量	掌握			
		3.2 变量及变量的作	PROG_F	变量	掌握			

	化程序设计的逻辑控制(包括顺序、选择、循环)；掌握各种控制结构中各语句的用法。	用域：变量的命名，变量的类型，变量之间类型转换，变量的作用域范围等。	D_03_02					
			PROG_FD_02_01	基本数据类型	掌握			
			PROG_FD_02_02	类型转换	掌握			
		3.3 Java 中的运算符和结合性：算术运算符、关系运算符、逻辑运算符、位运算符以及运算优先级和结合性的用法。	PROG_FD_03_03	内置运算	掌握			
			PROG_FD_03_04	表达式	掌握			
		3.4程序的流程控制：顺序结构，if 语句、switch语句引导的选择结构,for语句(for each)、while 语句引导的循环结构，break、continue 语句引导的跳转结构等。	PROG_FD_04	流程控制	掌握			
			PROG_FD_04_01	选择语句	掌握			
			PROG_FD_04_02	循环语句	掌握			
			PROG_FD_04_03	其他控制语句	掌握			
4. 数组和字符串	理解 Java 中数组和字符串与其它语言中相应概念的异同；掌握数组和字符串的使用方法。	4.1一维数组： 一维数组的声明和初始化，一维数组的内存分配，一维数组的引用，利用length属性来遍历整个数组，程序举例。	PROG_FD_06	数组	掌握	6	4	2
		4.2二维数组：二维数组的声明和初始化，二维数组的内存分配，二维数组的引用，利用length属性来遍历整个数组，程序举例。	PROG_FD_06	数组	掌握			
		4.3String类：String类的特点，基本使用方法，程序举例。	PROG_FD_09	字符串	综合应用			
		4.4StringBuffer类：StringBuffer类的特点，与String类的区别，基本使用方法，程序举例。	PROG_FD_09	字符串	综合应用			
		4.5字符串的特殊处	PROG_F	字符串	综合应			

		理方法：StringTokenizer 类的功能，基本使用方法，程序举例。	D_09		用			
5. 类与对象	掌握类的定义、创建、实例化；掌握对象的创建和使用； 掌握修饰符； 掌握对象间的赋值、对象的有效范围、构造函数； 理解方法是如何成为对象的对外接口的；会使用内部类；理解命令行参数的使用。	5.1类与对象：类的定义，对象的产生与使用，对象的比较，对象的生命周期，匿名对象，实现类的封装性。	PROG_00_01_01	类的概念和定义	掌握	10	6	4
			PROG_00_01_01_01	类的成员	掌握			
			PROG_00_01_01_02	成员的访问方式和权限	综合应用			
			PROG_00_01_01_03	函数/方法重载	掌握			
			PROG_00_01_02	对象的概念和定义	掌握			
		5.2构造函数：构造函数的定义与作用，构造函数的重载，构造函数的使用技巧。	PROG_00_01_03	对象的初始化和删除	综合应用			
		5.3this引用： this 引用的使用方法。	无		综合应用			
		5.4与垃圾回收有关的知识：finalize方法，System.gc()的使用。	PROG_00_01_03	对象的初始化和删除	综合应用			
		5.5static 关键字：静态变量，静态方法，静态变量的初始化方法，静态变量的用途。	无		综合应用			
		5.6内部类：内部类的定义，内部类的使用，内部类如何被外部引用，方法中定义内部类，内部类的名字等。	无		综合应用			
		5.7命令行参数	无		综合应用			
6类与对象的高	理解继承和多态的概念；掌握	6.1类的继承：继承的方式，继承的语法，	PROG_00_02_0	继承关系的定	综合应用	12	8	4

级特性	属性和方法在继承中的特点；掌握构造函数的继承和重载；理解包的概念，会使用包；类中成员的使用权限；掌握接口的概念和用法。	数据和方法的覆盖，super关键字的用法，运行时多态性，instanceof 运算符的用法等。	3_01	义				
			PROG_00_02_04	多态性				
		6.2final类和静态类	无					
		6.3包：包的语句及应用，import语句及应用，包与目录结构、环境变量之间的关系，父包与子包之间的关系，JDK中常用包。	无		综合应用			
		6.4类内成员的访问权限：公有、私有、保护、缺省（包内）。	PROG_00_01_01_02	成员的访问方式和权限	综合应用			
		6.5接口：接口的概念与定义，接口的作用，接口使用实例。	无		综合应用			
7. 异常处理	理解异常的概念；掌握异常控制机制；了解 Java 中的异常类；理解 Java 中的异常类的使用、异常的重新抛出；掌握自己定义异常。	7.1什么是异常	PROG_00_03_01	异常处理的基本思想	掌握	4	2	2
		7.2try … catch … finally 语句	PROG_00_03_02	异常处理的语法	综合应用			
		7.3异常的分类：JDK中异常的层次结构和分类及特点，常见异常。	无		综合应用			
		7.4throw 和 throws 的使用	PROG_00_03_02	异常处理的语法	综合应用			
		7.5定义自己的异常	无		综合应用			
8GUI 设计	掌握用Swing来设计图形用户界面的方法，尤其是组件、容器、布局管理器等概念；理解事件处理模型，掌握事件源、事件、事件处理者等概念，让程序能够响应用户	8.1图形化用户界面：组件和容器的概念，常用容器类，容器的布局管理器。	PROG_GN_02_01	容器的概念与定义	了解	10	6	4
		8.2事件处理模型：事件类，事件监听器，事件及其相应的监听器接口，事件适配器。	PROG_EV_01_01	监听与广播机制	综合应用			
			PROG_EV	事件	综合应用			

	的操作；了解Swing各个组件的用法及所采用的事件处理接口。		_01_02					
			PROG_EV_01_03	事件源与事件监听器	综合应用			
			PROG_EV_01_02	组件交互	综合应用			
		8.3 Swing组件库	PROG_EV_01_02	组件交互	综合应用			
9. 泛型与集合框架	掌握Collection的概念、ArrayList和LinkedList用法、HashMap的用法、迭代器的用法。	9.1 泛型：泛型类声明、使用泛型类声明对象。	PROG_GN_02_01	容器的概念与定义	掌握	6	4	2
		9.2 链表：ArrayList和LinkedList泛型类、常用方法、应用举例。	无		综合应用			
		9.3 散列映射：HashMap泛型类、常用方法、遍历散列映射。	无		综合应用			
		9.4 树集：TreeSet泛型类、常用方法、树映射、应用举例等。	无		综合应用			
		9.5 迭代器	PROG_GN_02_02	迭代器的概念与定义	综合应用			
10.输入输出流	掌握输入输出的总体结构；掌握流的概念；理解各种流的使用方法。	10.1I/O流概述	PROG_FD_10_01	输入/输出的基本概念和语法	掌握	6	4	2
		10.2文件处理：File、FileInputStream 、FileOutputStream、RamdomAccessFile类的使用。	PROG_FD_10_02	文件操作	综合应用			
		10.3节点流、过滤流和字符流：InputStream,OutputStream,BufferedInputStream,BufferedOutputStream ,Reader,Writer 类的使用。	无		综合应用			
11.多线程	掌握多线程的创建及启动；理解线程之间的数据交流；理解	11.1 进程与线程	无		掌握	4	2	2
		11.2 Java中的线程：主线程、线程的状态与生命周期、线程调	无		掌握			

	线程的同步；熟悉线程的调度。	度与优先级。						
		11.3Thread类与线程的创建：使用Thread的子类、使用Thread类、实现runable接口。	无		掌握			
		11.4线程的常用方法	无		掌握			
		11.5线程同步与线程联合	无		综合应用			
12.网络程序设计	掌握网络编程的基本概念；能编写简单的网络程序。	12.1网络编程基础知识：IP地址和端口号，UDP和TCP，socket等。	无		掌握	4	2	2
		12.2编写TCP网络程序：ServerSocket，Socket类及其使用方法，TCP服务器端程序和客户端程序。	无		综合应用			
	学时合计					72	46	26
说明：周4学时，共18周，总计72学时，理论和实践同步，机房上课，边讲边练。								