《计算与软件工程Ⅰ》

1.1 课程概述

1.1.1 课程目标与定位

《计算与软件工程Ⅰ》，又名《计算与软件工程——个人级软件开发》，课程在软件工程理念指导下，侧重于程序设计教学，以一个计算示例的迭代式增量开发实践为线索，全面培养学生在个人开发级别的小规模软件系统构建能力，让学生初步体验软件工程方法与技术在系统开发中的关键作用。具体教学内容包括：（1）程序设计基础，面向对象程序设计语言；（2）OOA、OOD、调试与测试等软件工程知识；（3）个人级别的软件开发活动管理，个人级别的软件职业知识。

通过本课程的学习，学生应该能够：

- 掌握程序设计的基本思想。
- 理解迭代式软件开发的基本过程。
- 掌握面向对象分析、设计、构造的基本思想，能够使用 OOA 和 OOD 的思想、熟练使用 OOPL 在个人级别进行小规模软件系统的构建。
- 理解封装思想，掌握类、包等基本概念，能够熟练使用类、接口等程序设计机制。掌握继承的思想，能够正确使用继承机制构建复杂类层次结构。掌握多态的思想，能够熟练使用接口等实现多态。
- 理解类库的概念和构造方法。了解基本的图形库和网络库。
- 掌握 UML 的基本概念和常用图（包括用例图、类图和顺序图），能够使用一种 UML 建模工具绘制表达软件分析与设计的简单图。
- 能够熟练使用一种 IDE 进行小规模程序的开发。
- 能够了解个人级别上的软件职业知识，按照个人软件过程的基本思想记录个人软件开发活动。

本课程是专业教学计划中一门关键入门课程，系统讲授软件工程方法指导下的程序设计，在本科一年级实施。可以在《计算系统基础》等导论类课程之后执行，也可以做为第一门专业课程执行。

本课程是《计算与软件工程》课程的第一部分，是《计算与软件工程Ⅱ/Ⅲ》的技术基础。本课程详细讲解软件工程原则指导下的程序设计，建立学生工程观指导下在个人级别构建小规模软件系统的综合能力。在《计算与软件工程Ⅱ》中将进一步深化学生对软件工程原则的理解以及合作构建中小规模软件系统的综合能力。

本课程全面介绍面向对象程序设计语言知识以及相应的面向对象分析与设计基础知识，为后继课程提供了面向对象软件开发的技术基础。同时，本课程初步建立学生的软件工程理念，在个人开发级别上教授学生软件工程基础知识和实践，包括调试、测试、集成以及基于个人软件过程思想的初步计划能力，这些能力使得学生在学习后继课程时具备初步的软件工程能力。

1.1.2 知识矩阵

表 2. 《计算与软件工程Ⅰ》课程知识矩阵

知识点	k/c/a	E/D/0	讲课时数	课外阅读课时
软件工程基础				
软件过程模型			1	1
迭代式软件开发	a	E		
软件开发技术				
软件需求			1	1
需求分析（用例图），用例文本描述	a	E		
软件设计			1	1
设计类图、顺序图	a	E		
软件构造实践			2	1
API 设计与使用（API 使用）	c	D		
代码重用与代码库	k	0		
调试/错误隔离技术	a	E		
代码规范	a	E		
软件测试			2	1
单元测试（初步）	a	E		
程序设计语言基础				
程序设计语言的历史			2	0
机器语言	k	E		
汇编语言	k	E		
结构化程序设计语言	c	E		
面向对象程序设计语言	c	E		
其它程序设计语言	k	0		
程序设计环境			3(1)	2(1)
计算机系统的基本概念	k	E**		
冯·诺依曼体系	k	E**		
操作系统与系统软件	k	E**		
汇编程序、编译器、解释器	k	0		
虚拟机工作原理			1	0
虚拟机的概念	c	D		
虚拟机层次机构	c	D		
中间语言	c	D		
不同机器上运行代码的安全性问题	k	0		
声明和类型			3(1)	2(0.5)
类型的值集和操作集的概念	a	E**		
声明类型（绑定、可见性、作用域与生存期）	a	E**		
类型检查概论	c	D		
抽象机制			3(1)	2(0.5)
过程和函数等抽象机制	a	E**		
参数化机制（值传递和引用传递）	a	E**		
活动记录和内存管理	k	D		

类型参数和参数化类型	k	D		
程序设计基础				
程序设计基本结构			9(3)	5(1)
变量、类型、表达式和语句	a	E**		
高级语言的基本语法和语义	a	E**		
输入和输出基础	a	E*		
顺序、条件和循环控制结构	a	E**		
函数定义、调用和参数传递	a	E*		
结构化分解技术	a	E*		
算法和问题求解			2	2
问题求解策略	a	E*		
问题求解算法	a	E*		
算法实现策略	a	E*		
算法的概念和特性	a	E*		
问题求解技术	a	E*		
基本数据结构			6(2)	5(1)
数组	a	E**		
字符串和字符串处理	a	E**		
数据在存储器中的表示	a	E*		
静态分配、栈式分配和堆式分配	a	E*		
运行时的存储器管理	a	E**		
指针和引用	a	E*		
递归			1	1
递归的概念	k	E*		
事件驱动程序设计			2	2
事件处理方法	c	E*		
事件传播	c	E*		
异常处理	c	E*		
面向对象程序设计（及 Java 语言实现）				
封装与信息隐藏			8	3
类与对象	a	E		
对象和方法的内部表示	a	E		
组合	a	E		
接口	a	E		
访问控制	a	E		
静态变量与方法	a	E		
类的初始化	a	E		
对象垃圾收集机制	a	E		
内部类	a	E		
异常处理	a	E		
继承			3	1
类的重用	a	E		
类的层次	a	E		
多态性			3	1

子类型多态和继承	a	E		
Java 高级语言特性			1	1
部署	a	D		
Java 常用类库			5	5
容器类	a	E		
String	a	E		
输入输出	a	E		
网络编程	a	D		
图形用户界面	a	E		
软件工程管理				
软件配置管理			1	1
版本管理（版本更新、提交）	a	E		
个体软件过程			1	2
PSP0 级	c	O		
度量（时间，规模）	c	O		
职业实践				
职业实践基础			2	1
职业道德与行为规范	k	D		
社会、法律、历史、职业的问题与利害关系	k	O		
软件的经济效益	k	O		
软件工程工具				
软件开发工具				16(12)
软件建模工具（可选 StartUml，Rational Rose，Enterprise Architecture 等）	c	D		
集成开发环境（建议 Eclipse）	a	E**		
软件测试工具（建议 JUnit）	a	E		
代码检查工具（建议 CheckStyle）	a	E		
项目管理工具				6
配置管理工具（建议 SVN 或 CVS）	a	E		
软件工程产物				
代码			1	1
代码	a	E		

- k:知识；c:理解；a:应用（Bloom 分类法）。
- E:核心知识点，必须包含在课程内容中；D:最好包含在课程中；O:选讲内容；
- E*与 E**:如果学生未学习过计算系统基础和程序设计基础，E*与 E**知识点按照栏内课时讲解；否则复习 E**知识点，课时数为括号内课时。

1.1.3 课程教学组织方式

如图 3 所示，《计算与软件工程Ⅰ》围绕一个具体计算系统示例的迭代式增量开发过程组织教学。

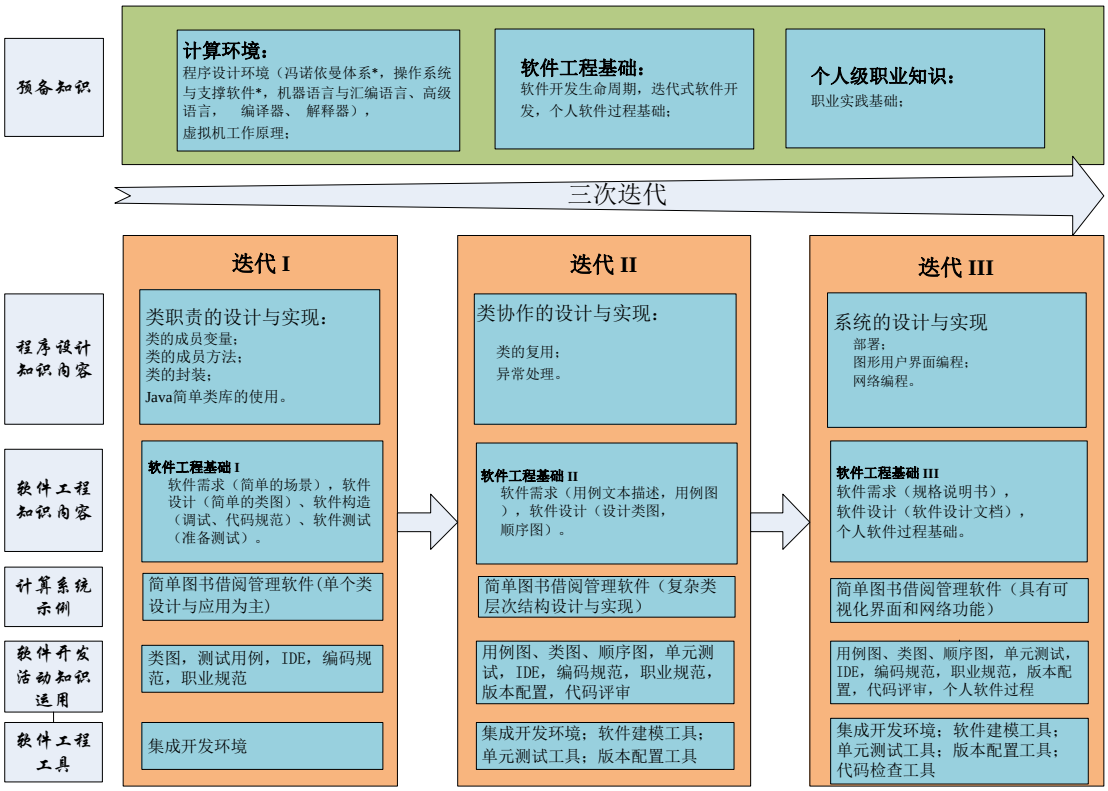

图 3. 《计算与软件工程Ⅰ》的教学组织示意图

在教学内容上，《计算与软件工程Ⅰ》融合了程序设计、个人开发级别的软件工程方法和职业素质培养，全面培养学生在个人开发级别上构建小规模软件系统的能力。传统程序设计课程往往偏重于技术的讲解，课程按照程序设计知识点展开。这种做法只能让学生了解分散的程序设计知识，学生很难从工程的角度了解软件构建过程。导致了学生割裂看待程序设计与软件工程，较难应用软件工程方法指导程序设计；以程序设计思维而不是工程思维来思考问题，遇到问题时先进行程序设计而不是寻找工程化解决方案；重程序设计技巧，轻软件工程方法，热衷于探索各种机巧的程序设计方法，忽视需求分析、评审、测试等软件工程的基本原则与方法。而本课程在每个教学阶段均坚持融合个人级软件工程、程序设计和基本职业素质教学，要求学生始终使用软件工程原则指导程序设计，为学生在专业学习的开始阶段建立起软件开发的工程观念，并建立起正确的基本职业素质认知。

在教学组织上，《计算与软件工程Ⅰ》以一个计算系统示例和一个学生实践用例的迭代式增量开发为线索，将个人开发级别的软件工程方法和程序设计教学相结合。迭代式开发也被称作迭代增量式开发或迭代进化式开发，在迭代式开发方法中，整个开发工作被组织为一系列的短小的项目，被称为一系列的迭代。采用这种方法，开发工作可以在需求被完整地确定之前启动，并在一次迭代中完成系统的一部分功能或业务逻辑的开发工作。再通过客户的反馈来细化需求，并开始新一轮的迭代。为了培养学生能够从工

程的角度看待软件开发，本课程设计了一个案例，分为三次迭代，为学生完整展示小规模通用软件系统构建过程。**本课程依据计算系统示例与学生实践用例的迭代开发过程，循序渐进、由浅入深组织程序设计知识与相关软件工程知识的融合教学。**学生可以同时学习到相应的程序设计和软件工程原则，并在教师的引导下体会这些原则在实际软件开发中的应用。软件工程是一门实践性非常强的学科，那种认为先介绍知识，然后附加一两个实践练习的教学方式割裂了知识与实践，无法帮助学生深入理解软件工程思想。《计算与软件工程Ⅰ》将程序设计、软件工程知识教学结合案例教学同步进行，按照软件工程原则安排每个迭代的分析、设计、程序设计和测试实践。学生在学习了相应知识后，立刻就可以看到它们的实践应用，获得及时反馈，加深对软件工程原则的理解。

《计算与软件工程Ⅰ》将计算示例分成三次迭代：

- 第一次迭代：基于类职责的设计与实现，利用软件工程和程序设计知识构建系统；
- 第二次迭代：基于类协作的设计与实现，利用软件工程和程序设计知识二次构建系统；
- 第三次迭代：基于完整系统的设计与实现，利用 GUI 与网络等，进行三次增量开发。

课程选择 Java 作为编程语言进行教学。Java 作为当前业界最为广泛使用的编程语言，提供了面向对象程序设计的全面机制，适合作为入门语言选用，也是后继课程教学的语言基础之一。

1.2 课程内容

《计算与软件工程Ⅰ》课程首先介绍程序设计背景知识，为课程的展开提供基础。然后，课程将内容依照三次迭代进行组织，每个迭代教学内容均包括程序设计、软件工程基础和计算系统示例，其中计算系统示例在三个阶段是按照迭代式开发组织的。在课程内容设计过程中，知识教授和计算系统示例的迭代开发内容相匹配是一个重要的原则。

根据学生是否学习过《计算系统基础》或《程序设计基础》等前导性知识课程，课程的课时安排如表 3 与表 4 所示：

表 3. 《计算与软件工程Ⅰ》课时安排（作为入门课程）

	课堂讲授	课后实验	课后阅读/作业
预备知识	3	0	4
计算系统示例说明	1	0	0
迭代一	25	24	23
迭代二	20	20	23
迭代三	15	20	14
总计	64	64	64

表 4. 《计算与软件工程Ⅰ》课时安排（已经学习过计算系统等基础课程）

	课堂讲授	课后实验	课后阅读/作业
预备知识	1	0	2
计算系统示例说明	1	0	0
迭代一	11	8	9
迭代二	20	20	23
迭代三	15	20	14
总计	48	48	48

1.2.1 预备知识

本部分讲解计算基础、软件工程基础、个人级软件职业基础等预备知识，为学生学习个人级别的小规模软件开发做好准备。教学内容具体包括：

- 计算环境：程序设计环境（冯•诺依曼体系*，操作系统与支撑软件*，机器语言与汇编语言、高级语言、编译器、解释器），虚拟机工作原理；（如果未学习过《计算系统基础》《程序设计基础》等入门课程，*内容必须详细讲解，否则可以忽略）
- 软件工程基础：软件开发生命周期，迭代式软件开发，个人软件过程基础；
- 个人级职业知识：职业实践基础。

1.2.2 计算系统示例与学生实践用例说明

一、计算系统示例（教学讲解用）：图书借阅系统

“图书借阅系统”是一个具有图形用户界面的基于网络的用于在大学中管理图书借阅的软件。软件功能如下：

- 管理员：
  - 图书管理：图书分为普通图书和珍本图书，可以进行增删改的操作；
  - 借阅人管理：借阅人分为本科生、研究生、教职工，可以进行增删改的操作。
- 学生用户：
  - 查询图书；
  - 根据自身权限借阅图书，并可以续借图书；
  - 归还图书；
  - 接近图书借阅期限时，可以获得相应提醒。
- 教师用户：
  - 具备学生用户功能；
  - 可以要求学生借阅者提前归还图书。
- 网络同步：
  - 可以和服务器进行同步，用户在服务器上设定用户名和密码，登录后可以进行同步；
  - 同一时刻，仅允许同一个用户的一个客户端登录。
- 系统其它假定：
  - 每种图书只有一本；

- 不使用数据库。

二、计算系统实践用例（学生实验用）：大学公用场所资源预订管理系统

“大学公用场所资源预订管理系统”是一个具有图形用户界面的基于网络的用于在大学中管理公用场所资源预订的软件。软件功能如下：

- 管理员：
  - 帐号管理：可以对帐号进行增删改操作；
  - 公用场所资源管理：可以对房间资源进行增删改操作，资源包括教室、体育场馆、会议厅等。
- 一般用户：
  - 查询场地在某一时段的使用情况；
  - 可以预定体育场馆。
- 中级用户（教师，团委学生会）：
  - 预订体育场馆；
  - 预订会议室。
- 高级用户（教务处）：
  - 查询公用场所在某一时段的使用情况；
  - 预订所有公用场所；
  - 有调整其他用户预订的权力。
- 系统其它假定：
  - 不使用数据库。

1.2.3 迭代一

学习如何构建一个基于命令行的简单图书借阅管理软件。教学内容具体包括：

- 程序设计知识内容：
  - 类的成员变量
    - 名字（名字，命名规则）；
    - 变量与类型（基本数据类型，变量，类型检测，命名常量）；
    - 表达式与赋值语句（赋值语句，算术表达式，类型转换，关系表达式和布尔表达式，短路求值，混合模式赋值）；
    - 其它常用数据类型（字符串类型，数组类型，记录类型与简单类定义，指针类型与引用类型）；
  - 类的成员方法
    - 语句层次的控制结构（条件语句，分支语句，循环语句，Goto 语句讨论）；
    - 子程序（子程序的基本原理，子程序的设计问题，参数传递方法，作用域与生存期，全局变量，函数副作用）；
    - 递归；
  - 类的封装
    - 封装的基本概念

        - 对象初始化和清理 I
        - 访问控制 I
        - 静态变量与方法
    - Java 简单类库的使用
        - String
        - 容器类
        - 输入输出文件访问
    - 注：如果未学习过《计算系统基础》《程序设计基础》等入门课程，*内容必须详细讲解，否则简单复习基础知识，重在认知 Java 语法。
- 软件工程知识内容：
    - 软件需求（简单的场景）；
    - 软件设计（简单的类图）；
    - 软件构造（调试、代码规范）；
    - 软件测试（准备测试）。
- 计算系统示例知识运用要求：设计实现一个简单图书借阅管理软件。
    - 设计一个图书类，该类包含属性：图书名称，图书编号，是否为珍本图书，是否已经被借出，图书借阅人，归还时间；
    - 设计一个借阅人类，该类包含属性：姓名，ID，email，已借阅图书等；
    - 以上两个类针对每种属性具有相应读取和设置方法；
    - 采用集合类来组织对象；
    - 使用文本文件进行存取操作；
    - 设计一个静态类，使用常量表示人员的借阅时长。
- 软件开发活动知识运用要求：
    - 给出单个类的类图；
    - 给出测试用例；
    - 使用集成化开发环境（IDE）进行程序编写和调试；
    - 遵循代码规范；
    - 遵循个人职业规范。
- 软件工程工具：
    - 集成开发环境；
- 配套实验：设计一个简单的“大学公用场所资源预订管理系统”
    - 设计预定类，该类包含属性：预订时间、预定人员；
    - 设计一个场所类，该类包含属性：房间号、房屋可容纳人员数目、所属建筑、是否可用、预定列表（一个预定类的集合）；
    - 设计普通用户类。该类包含姓名、ID 号、所属院系或学校机关、手机等属性，具有预订方法；
    - 以上类针对每种属性具有相应读取和设置方法；
    - 采用集合类来组织对象；
    - 使用文本文件进行存取操作。
- 书面作业与上机作业：
    - *表达式运算（书面作业）；

- *循环、迭代语句使用（书面作业）；
- *函数使用（书面作业）；
- *数组的使用（书面作业）；
- *字符串使用（书面作业）；
- 对象初始化（书面作业）；
- 使用 IDE 编写并调试一个判断闰年，计算任意两个日期之间的天数的简单程序（上机作业）；
- 注：如果未学习过《计算系统基础》《程序设计基础》等入门课程，*内容为必须作业，否则为可选作业。

1.2.4 迭代二

学习如何构建一个基于命令行的图书借阅管理软件。教学内容具体包括：

- 程序设计知识内容：
  - 类的协作（继承，聚合，接口，多态，对象初始化和清理 II，访问控制 II，内部类，类的层次）；
  - 异常处理（异常处理概述，Java 中的异常处理）。
- 软件工程知识内容：
  - 软件需求（用例文本描述，用例图）；
  - 软件设计（设计类图，顺序图）。
- 计算系统示例知识运用要求：图书借阅管理软件（在迭代一的基础上进行开发，增加新的功能）。
  - Student 和 Staff 类继承借阅人类，Undergraduate 和 Graduate 类继承 Student 类；Staff 类中增加 request 方法，可以要求学生在 request 日期的 7 天内归还图书；Graduate 和 Staff 可以借阅珍本图书；
  - 简单登录，区分用户类型；
  - 管理员可以通过命令行方式增加、删除和修改图书；
  - 管理员可以通过命令行方式增加、删除和修改借阅人信息；
  - 管理员可以通过命令行方式列出某个借阅人所借的所有图书；
  - 借阅人可以通过命令行方式借阅图书，续借图书，归还图书；
  - 借阅人可以通过命令行方式列出某本图书的所有借阅人；
  - Staff 用户可以通过命令行方式要求学生用户提前归还图书。
- 软件开发活动知识运用要求：
  - 设计用例图、类图、顺序图；
  - 使用测试策略进行单元测试和多个类协同测试；
  - 使用集成化开发环境（IDE）进行程序编写和调试；
  - 使用版本配置工具进行代码版本管理；
  - 尝试计划个人工作，记录工作数据；
  - 遵循代码规范，进行代码评审；
  - 遵循个人职业规范。
- 软件工程工具：

- 集成开发环境；
- 软件建模工具；
- 单元测试工具；
- 版本配置工具。

- 配套实验：设计一个简单的“大学公用场所资源预订管理系统”（在迭代一配套实验的基础上进行开发，增加新的功能）
  - 设计教室、活动房间类，应当继承迭代一的房间类；教室类中增加多媒体设备属性；活动房间类增加功能说明属性；体育活动室和报告厅继承活动房间类；体育活动室增加预订价格属性；报告厅增加灯光音响属性；
  - 设计中级用户类（教师，团委，学生会），该类继承普通用户类；
  - 设计高级用户类（用户包括教务处），该类继承中级用户类；
  - 简单登录，区分用户类型；
  - 管理员可以通过命令行方式增加、删除、修改普通用户、中级用户和高级用户；可以通过命令行方式增加、删除、修改场所资源；
  - 用户可以通过命令行列出所有房间或某个房间在某个时间段的预定情况；
  - 普通用户可以通过命令行预订运动场所；
  - 中级用户可以通过命令行预订运动场所和会议室；
  - 高级用户可以通过命令行预订所有场所，同时可以调整所有用户的预订。
- 书面作业与上机作业：
  - 内部类的概念（书面作业）；
  - 写一个简单的图形系统，实现多态（上机作业）；
  - 实现输入输出的异常处理（上机作业）。

1.2.5 迭代三

学习如何构建一个基于图形用户界面的图书借阅管理软件。教学内容具体包括：

- 程序设计知识内容：
  - 部署；
  - 图形用户界面编程；
  - 网络编程；
- 软件工程知识内容：
  - 软件需求（规格说明书）；
  - 软件设计（软件设计文档）；
  - 个人软件过程基础。
- 计算系统示例知识运用要求：图书借阅管理软件（具有可视化界面和网络功能，在迭代二的基础上进行开发，增加新的功能）。
  - 图形用户界面：
    - 学生用户界面：可以显示自己已经借阅的图书已经预定归还日期；查询图书并借阅（可能由于借阅本数限制或该书以被别人借阅而失败）；
    - 教师用户界面：可以显示自己已经借阅的图书已经预定归还日期；查询图书并借阅（可能由于借阅本数限制或该书以被别人借阅而失败）

或要求学生用户提前归还；
    - 管理员界面：增加、删除或修改图书信息或借阅人信息。
  - 网络同步功能（使用 sockets 进行通信）：
    - 登录机制，登录后根据用户角色显示界面；
    - 锁机制。任意时刻仅允许一个用户进行操作；
    - 采用简单文件同步机制：每次程序开始时检查服务器版本时间是否比本机新，如果是更新本地文件；对服务器端文件加锁；文件退出时，将文件上传到服务器，将服务器端文件解锁。
- 软件开发活动知识运用要求：
  - 设计用例图、类图、顺序图；
  - 使用测试策略进行单元测试和多个类协同测试；
  - 使用集成化开发环境（IDE）进行程序编写和调试；
  - 使用版本配置工具进行代码版本管理；
  - 进行集成；
  - 按照估算、计划、跟踪、总结的个人软件过程实践进行软件开发活动，并记录相应工作数据；
  - 遵循代码规范，进行代码评审；
  - 遵循个人职业规范。
- 软件工程工具：
  - 集成开发环境；
  - 软件建模工具；
  - 单元测试工具；
  - 版本配置工具；
  - 代码检查工具。
- 配套实验：设计一个简单的“大学公用场所资源预订管理系统”（在迭代二配套实验的基础上进行开发，增加新的功能）。
  - 图形用户界面：
    - 普通用户界面：可以看到某时段所有场所预定情况；可以查询某时段特定场所预定情况；可以预定体育场馆；
    - 中级用户界面：可以看到某时段所有场所预定情况；可以查询某时段特定场所预定情况；可以预定体育场馆和会议室；
    - 高级用户界面：可以看到某时段所有场所预定情况；可以查询某时段特定场所预定情况；可以预定所有房间。
  - 网络功能（使用 sockets 进行通信）：
    - 登录机制，登录后根据用户角色显示界面；
    - 锁机制。任意时刻仅允许一个用户进行操作；
    - 采用简单文件同步机制：每次用户登录开始时检查服务器版本时间是否比本机新，如果是更新本地文件；对服务器端文件加锁；文件退出时，将文件上传到服务器，将服务器端文件解锁；
    - 高级用户调整结果后，原用户登录时可以得到提醒。
- 书面作业与上机作业：

- 网络编程概念（书面作业）；
- 使用复杂 Swing 类构建 GUI，例如：JTree（上机作业）；
- 使用 sockets 编写一个可以互相发送消息的程序（上机作业）。

1.3教学重点与难点以及解决方案

课程重点是使得学生在软件工程方法指导下，**建立个人开发级别上的构建小规模软件系统的能力**具体包括：

- 面向对象程序设计的知识与能力；
- 基础面向对象分析、设计、调试、测试等软件工程方法和能力；
- 基于个人软件过程的初步计划能力；
- 基本个人职业素质。

课程难点及解决方法如下：

- 如何有效解决割裂学习程序设计与软件工程方法的问题。

传统课程通常先介绍程序设计，然后再介绍软件工程方法，往往导致学生割裂认识这两种知识。课程通过一个贯穿课程的计算示例系统，将程序设计知识、软件工程方法融合在示例讲解中传授给学生，使得学生在示例学习中体会软件工程方法指导下的个人软件开发。

- 如何使学生在编程实践中考虑编码规范。

在课程讲解中描述编码规范的意义，并给出 SUN 公司的基本编码规范作为样例。在实验和作业中强调编码规范的实施，并使用代码静态检查工具 CheckStyle 对学生代码进行检查，将结果反馈给学生。

- 如何使学生尽快掌握基本单元测试技术。

在课程讲解中描述单元测试的意义，并讲解 Java 单元测试工具 JUnit 的具体使用，要求学生在实验和作用中提供单元测试案例代码。

1.4课程考核规则

本课程要求学生在掌握课程知识内容的同时，具有较强的实践能力，建议本课程考试分为笔试和机考两部分，同时计算系统实验考察得分应该计入最终总分。

- 笔试：采用闭卷形式，重点考察学生对基本概念和基本方法的掌握程度。建议题型为选择（考察基本概念的掌握程度）、阅读程序写出运行结果（考察跟踪程序运行能力）、程序改错（考察分析程序错误能力）和编写程序（考察程序设计能力）。
- 机考：在规定时间内，根据给定的题目设计程序，并最终调试运行。

期末总评成绩：笔试×50%＋机考×20%+实验×30%。