**《编译原理》课程教学大纲**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 英文课程名 | The Principle of Compiler | | 总 学 时 | | 48 | 学 分 | 3 |
| 课程编码 | G126013 | | 理论教学学时 | | 40 | 适用专业 | 软件工程 |
| 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 | 8 | 先修课程 | C++程序设计,离散数学,数据结构,操作系统原理,算法分析与设计 |
| 大类基础课程 |  | 上机学时 | 0 | 开课学院（部） | 计算机科学与技术学院 软件学院 |
| 专业课程 | 必修 | 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**

此课程针对软件工程专业的专业必修课，较为详细地解析了编译过程包括词法分析、语法分析、语义分析、中间代码生成、中间代码优化、目标代码生成与目标代码优化等编译步骤的关键技术。它是计算机形式化理论的入门，也为计算机语言设计提供必要的理论基础。该课程是一门综合课程，不仅需要离散数学、高级程序语言、操作系统的相关知识，还需要学生了解数据结构和算法设计与分析的基本知识；同时，本课程也能承接软件工程、软件项目管理、软件形式化方法、.Net技术、Java EE技术、移动应用开发等课程。主要培养学生综合运用计算机相关理论和技术，设计复杂算法以及设计复杂大型系统软件的能力。课程突出学生知识、能力、素质的协调发展，注重学生学习能力培养。

**二、教学目标**

**2.1 课程教学目标**

课程主要教学目标是培养学生将较为透彻地理解编译器的概念及原理，掌握编译器中常用的主要算法和技巧，并具有设计实现简单程序语言的能力。通过课堂教学、分组讨论、课后自学等方式的教学过程，努力培养学生具有形式化分析问题的能力，能够将具体的计算问题利用形式化工具建模，从而设计出可求解该计算问题的算法和程序。同时，通过学习构造和实现编译程序的过程和原理，为学生理解程序的正确运行和写出高质量代码打下坚实基础。课程要求掌握文法/正规式/自动机、词法分析、语法分析、语义分析、中间代码生成、中间代码优化、目标代码生成与目标代码优化的基本原理，为学习和深刻理解计算机专业各后续课程做好必要的知识准备。进一步提高学生的抽象思维、逻辑推理能力和算法设计能力，为从事计算机的应用提供坚实的理论基础。通过课程的学习，要求学生能够达到对概念和方法透彻理解的程度。通过对课程知识点和其演进历史的学习，形成严谨的治学、行事态度和勇于创新、锐意进取的价值取向。

课程的具体目标为：

目标1：掌握文法基础、有限自动机及其简化、正规式、First集与Follow集与Select集、表驱动LL(1)分析、移进与规约、LR(0)分析、LR(1)分析、属性文法与翻译模式概念、树遍历属性计算、语义分析概念、拉链与代码回填、活动记录组织、过程调用机制、基本块与流图。

目标2：熟悉文法二义性、正规文法与正规式的等价、正规式与有限自动机的等价、词法分析概念、词法分析程序的设计与实现、消除左递归、消除左公因子、SLR(1)分析、LALR(1)分析、继承属性与综合属性、常用中间表示方式、布尔表达式语法制导翻译、存储分配策略、参数传递机制、数据流分析、基本块的DAG表示。

目标3：理解编译过程各步骤与衔接、文法分类、正规文法与有限自动机的等价、词法分析程序自动构造与Lex、递归下降LL(1)分析、S-属性文法和L-属性文法、S-翻译模式和L-翻译模式、静态语义检查、典型语句中间表示、运行时存储组织的作用与任务、程序运行时存储空间的布局、参数传递机制、目标代码生成、目标代码优化。

**2.2 课程目标与毕业要求（指标点）对应关系**

本课程所支撑的毕业要求及其细分指标点如下：

【毕业要求3】设计/开发解决方案：能够设计针对复杂软件工程问题的解决方案，设计满足特定需求的系统、单元（部件）或工艺流程，并能够在设计环节中体现创新意识，考虑社会、健康、安全、法律、文化以及环境等因素。

指标点3.2：能够对复杂软件工程问题进行需求分析，能够运用软件工程思想构建系统总体解决方案，确定设计目标并进行模块分解。

指标点3.3：按照工程化要求对软件系统的子模块、子单元或部件进行设计。

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 | 实验 | 作业 | 课堂测验 | 课堂讨论 |
| 目标1：掌握文法基础、有限自动机及其简化、正规式、First集与Follow集与Select集、表驱动LL(1)分析、移进与规约、LR(0)分析、LR(1)分析、属性文法与翻译模式概念、树遍历属性计算、语义分析概念、拉链与代码回填、活动记录组织、过程调用机制、基本块与流图。 | 指标点3.3：按照工程化要求对软件系统的子模块、子单元或部件进行设计。 | √ | √ | √ |  | √ |
| 目标2：熟悉文法二义性、正规文法与正规式的等价、正规式与有限自动机的等价、词法分析概念、词法分析程序的设计与实现、消除左递归、消除左公因子、SLR(1)分析、LALR(1)分析、继承属性与综合属性、常用中间表示方式、布尔表达式语法制导翻译、存储分配策略、参数传递机制、数据流分析、基本块的DAG表示。 | 指标点3.2：能够对复杂软件工程问题进行需求分析，能够运用软件工程思想构建系统总体解决方案，确定设计目标并进行模块分解。  指标点3.3：按照工程化要求对软件系统的子模块、子单元或部件进行设计。 | √ |  | √ |  | √ |
| 目标3：理解编译过程各步骤与衔接、文法分类、正规文法与有限自动机的等价、词法分析程序自动构造与Lex、递归下降LL(1)分析、S-属性文法和L-属性文法、S-翻译模式和L-翻译模式、静态语义检查、典型语句中间表示、运行时存储组织的作用与任务、程序运行时存储空间的布局、参数传递机制、目标代码生成、目标代码优化。 | 指标点3.2：能够对复杂软件工程问题进行需求分析，能够运用软件工程思想构建系统总体解决方案，确定设计目标并进行模块分解。  指标点3.3：按照工程化要求对软件系统的子模块、子单元或部件进行设计。 | √ | √ | √ |  | √ |

**三、课程教学内容及学时分配**

**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 | 绪论 | 1. 什么是编译程序 2. 编译过程概述 3. 编译程序的结构 4. 课程要求 | 2 | 教学要求：   1. 掌握编译程序的概念； 2. 理解编译程序的组成； 3. 理解编译程序的逻辑结构； 4. 了解编译程序的伙伴程序。   教学重点：   1. 编译程序的概念； 2. 编译程序的组成。   教学难点：编译过程概述。  教学方法：讲授+课堂讨论 | 1. 编译程序的组成。 | 自学：   1. 编译技术发展的历史。   讨论：   1. 编译与解释的关系。 |
| 2 | 文法与自动机 | 1. 文法  2. 有限自动机的概念及化简算法  3. 正规式  4. 正规式与有限自动机的转换  5. 正规文法与有限自动机的转换 | 8 | 教学要求：   1. 掌握文法基础； 2. 理解文法分类； 3. 熟悉文法二义性； 4. 掌握有限自动机及其简化； 5. 掌握正规式； 6. 熟悉正规式与有限自动机的等价； 7. 理解正规文法与有限自动机的等价。   教学重点：   1. 非确定有限自动机转换为确定有限自动机； 2. 确定有限自动机的简化； 3. 正规式转换为有限自动机。   教学难点：文法二义性、文法分类、非确定有限自动机转换为确定有限自动机、确定有限自动机的简化。  教学方法：讲授+实例练习+参与式教学 | 1. 给出文法写语言； 2. 给出语义写文法； 3. 判断文法二义性； 4. 推导、归约与语法树。 | 自学：   1. 正规文法与有限自动机的等价。   讨论：   1. 非确定有限自动机转换为确定有限自动机； 2. 确定有限自动机的简化； 3. 正规式转换为有限自动机。 |
| 3 | 词法分析 | 1. 词法分析概念  2. 词法分析程序的设计和实现  3. 词法分析程序的自动构造与LEX简介 | 2 | 教学要求：   1. 熟悉词法分析概念； 2. 熟悉词法分析程序的设计和实现； 3. 理解词法分析程序的自动构造和Lex。   教学重点：   1. 词法分析程序的设计和实现。   教学难点：词法分析程序的自动构造。  教学方法：讲授+实例练习+参与式教学 | 1. 对给出代码写出单词流。 | 自学：   1. 词法分析程序的自动构造与LEX简介。   讨论：   1. 词法分析中专题的设计方案。 |
| 4 | 自顶向下的语法分析 | 1. First集与Follow集与Select集  2. LL(1)文法  3. 消除左递归与消除左公因子 | 4 | 教学要求：   1. 掌握First集与Follow集与Select集； 2. 理解递归下降LL(1)； 3. 掌握表驱动LL(1)； 4. 熟悉消除左递归； 5. 熟悉消除左公因子。   教学重点：   1. First集与Follow集与Select集； 2. 表驱动LL(1)。   教学难点：消除左递归、递归下降LL(1)。  教学方法：讲授+实例练习+参与式教学 | 1. First集与Follow集与Select集计算； 2. LL(1)文法判断； 3. 表驱动LL(1)； 4. 递归下降LL(1)。 | 自学：   1. 递归下降LL(1)； 2. 消除左公因子。   讨论：   1. 递归下降LL(1)； 2. 表驱动LL(1)； 3. 消除左递归。 |
| 5 | 自底向上的语法分析 | 1. 移进与规约  2. LR（0）分析  3. SLR（l）分析  4. LR（1）分析  5. LALR（1）分析 | 6 | 教学要求：   1. 掌握移进与规约； 2. 掌握LR（0）分析； 3. 熟悉SLR（l）分析； 4. 掌握LR（1）分析； 5. 熟悉LALR（1）分析。   教学重点：   1. LR（0）分析； 2. LR（1）分析。   教学难点：句柄与活前缀、LR（0）和LR（1）有限自动机构造。  教学方法：讲授+实例练习+参与式教学 | 1. LR（0）分析； 2. SLR（l）分析； 3. LR（1）分析； 4. LALR（1）分析。 | 自学：   1. 自底向上分析的一般过程和问题； 2. LALR（1）分析。   讨论：   1. LR（0）分析； 2. LR（1）分析。 |
| 6 | 语法制导的语义计算 | 1. 属性文法和翻译模式的概述  2. 属性文法语义计算  3. 翻译模式语义计算 | 4 | 教学要求：   1. 掌握属性文法和翻译模式概念； 2. 熟悉继承属性与综合属性； 3. 掌握树遍历属性计算； 4. 理解S-属性文法和L-属性文法； 5. 理解S-翻译模式和L-翻译模式。   教学重点：   1. 树遍历属性计算； 2. L-翻译模式。   教学难点：树遍历属性计算。  教学方法：讲授+实例练习+参与式教学 | 1. 根据功能设计属性文法； 2. 根据功能设计翻译文法； 3. 树遍历属性计算。 | 自学：   1. 继承属性与综合属性； 2. S-翻译模式。   讨论：   1. L-属性文法； 2. L-翻译模式。 |
| 7 | 语义分析与中间代码生成 | 1. 语义分析概念  2. 静态类型检查及实例  3. 常用中间表示方式  4. 典型语句中间表示  5. 布尔表达式翻译  6. 拉链与回填技术 | 6 | 教学要求：   1. 掌握语义分析概念； 2. 理解静态类型检查及实例； 3. 熟悉常用中间表示方式； 4. 理解典型语句中间表示； 5. 熟悉布尔表达式翻译； 6. 掌握拉链与回填技术。   教学重点：   1. 布尔表达式翻译； 2. 拉链与回填技术。   教学难点：类型系统、拉链与回填技术。  教学方法：讲授+实例练习+参与式教学 | 1. 布尔表达式翻译； 2. 拉链与回填技术。 | 自学：   1. 常用中间表示方式； 2. 静态类型检查实例。   讨论：   1. 布尔表达式翻译； 2. 拉链与回填技术。 |
| 8 | 运行时存储空间的组织 | 1. 运行时存储组织的作用与任务  2. 程序运行时存储空间的布局  3. 存储分配策略  4. 活动记录  5. 过程调用和参数传递 | 4 | 教学要求：   1. 理解运行时存储组织的作用与任务； 2. 理解程序运行时存储空间的布局； 3. 熟悉存储分配策略； 4. 掌握活动记录的组织； 5. 掌握过程调用机制； 6. 熟悉参数传递机制。   教学重点：   1. 活动记录； 2. Display表； 3. 静态链与动态链。   教学难点：Display表。  教学方法：讲授+实例练习+参与式教学 | 1. 活动记录的栈存储； 2. Display表应用； 3. 静态链应用。 | 自学：   1. 存储分配策略； 2. 参数传递机制。   讨论：   1. 活动记录； 2. Display表。 |
| 9 | 代码生成与代码优化 | 1. 基本块、流图和循环  2. 数据流分析  3. 基本块的DAG表示  4. 目标代码生成  5. 目标代码优化 | 4 | 教学要求：   1. 掌握基本块、流图和循环； 2. 熟悉数据流分析； 3. 熟悉基本块的DAG表示； 4. 理解目标代码生成； 5. 理解目标代码优化。   教学重点：   1. 基本块、流图和循环。   教学难点：数据流、基本块的DAG表示。  教学方法：讲授+课堂讨论 | 1. 基本块、流图和循环。 | 自学：   1. 基本块的DAG表示； 2. 目标代码优化。   讨论：   1. 基本块、流图和循环。 |

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 | 正规式转换为有限自动机 | 2 | 设计 | 1-3 | 教学重点：   1. 编写将正规式转化为等价有限自动机的程序   难点：   1. 正规式到有限自动机的局部转换与整体转换的衔接 2. 有限自动机数据结构   教学方法：分组自学讨论 |  | 自学：   1. 正规式转化为等价有限自动机的方法   讨论：   1. 正规式到有限自动机的局部转换与整体转换的衔接 |
| 2 | 词法分析 | 2 | 设计 | 1-3 | 教学重点：   1. 编写根据有限自动机完成词法分析的程序   难点：   1. 对典型代码设计有限自动机   教学方法：分组自学讨论 |  | 自学：   1. 词法分析程序的自动构造与LEX简介。   讨论：   1. 词法分析中专题的设计方案。 |
| 3 | LL(1)分析表实现 | 2 | 设计 | 1-3 | 教学重点：   1. 编写由LL(1)分析表实现语法分析的程序   难点：   1. 预测分析表的实现 2. Select集的算法   教学方法：分组自学讨论 |  | 自学：   1. Select集的算法。   讨论：   1. 语法分析中错误提示方案。 |
| 4 | LR(0)分析表实现 | 2 | 设计 | 1-3 | 教学重点：   1. 编写由LR(0)完成语法分析的程序   难点：   1. LR(0)活前缀有限自动机的实现 2. Action表GOTO表的实现   教学方法：分组自学讨论 |  | 自学：   1. LR(0)活前缀有限自动机的程序实现。   讨论：   1. Action表GOTO表的程序实现。 |

**四、考核方式及成绩评定方式**

该课程的考核强调过程性考核。其总成绩分为进程性成绩和期末考试成绩两大部分，分别占50%。进程性成绩主要考核学生的作业、实验动手能力和课堂讨论表现三个方面。各个环节所占比例如下：

（1）期末考试：占总成绩的50%。试卷难度适中，填空、选择、判断、名词解释、简答等基础性题目的比例不超过30%，加大综合性题目的比例，重在考查学生运用微机接口知识分析和解决复杂工程问题的能力。

（2）作业：占总成绩的20%。教师针对某些知识模块布置一定数量的课后作业或课外思考题，以巩固知识或拓展思维。

（3）实验：占总成绩的20%。课程共设置2-4次课内实验，要求学生通过课外自学、文献与资料查阅等环节，综合运用计算机科学和工程技术完成实验设计、实施，并对实验结果进行分析。

（4）课堂讨论：占总成绩的10%。重点考察学生的自主学习能力、团队协作能力、语言表达与沟通能力。

期末考试采用“一页开卷”形式，以便将学生从死记硬背中解脱出来，从而将更多的精力放在知识的理解和灵活运用上，期末的试题主要是考察对基本原理的综合运用能力，以便能真正考核出学生的实际水平。

**五、教材、课程网址及参考书目**

教 材：编译原理，蒋宗礼、姜守旭，高等教育出版社，2017.08，第二版

课程在学校网络教学平台的地址(核心课程必填)：

参考书：【1】编译原理，张素琴, 吕映芝等，清华大学出版社，2015.06，第三版

【2】程序设计语言编译原理，陈火旺、刘春林等，国防工业出版社，2014.12，第三版

【3】Compilers: Principles, Techniques, and Tools，Alfred V. Aho 等，机械工业出版社，2011.01，第二版

**执笔者：张 端**

**审核者：田贤忠**

**课程教学团队成员：吕慧强、陈敏智、程振波、夏列刚、张 端**