**编译原理课程教学大纲**

1. **课程名称： 编译原理**
2. **课程代码:**  CST31110
3. **课程英文名称： Compiler Principles**
4. **课程负责人：** 张敏
5. **学时与学分**： 64**学时** 3.5**学分**
6. **课程性质：** 必修/选修
7. **课程类型:** 非限制选修
8. **适用专业**： 计算机科学与技术/信息安全/物联网工程
9. **选课对象**： 本科生
10. **预修课程**：数据结构、操作系统、程序设计基础、计算机组成原理、算法分析与设计
11. **使用教材：**

# 李文生，《编译原理与技术（第二版）》，清华大学出版社，出版时间2015年

1. Alfred V.Aho 等，《编译原理第2版（本科教学版）》，机械工程出版社，出版时间2009年5月

## 教材选用说明：国内新出版的通用教材，去除了许多陈旧落后的概念和要点，更新了很多新实例和知识点，理论与实践并重，加强实践环节。内容齐全，讲解深入，例题丰富，便于自学。其中【1】为主要教材，【2】为经典教材作为课外辅助学习资料。

1. **参考书目：**
2. 张素琴等编，《编译原理（第2版）》，清华大学出版社，出版时间2011年7月；
3. Kenneth C.Louden等编，《Compilers Construction: Principles and Practice》，机械工业出版社，出版时间2005 年 1 月
4. 何炎祥编，《编译原理（第三版）》，华中科技大学出版社，出版时间2010年8月
5. **开课单位**： 计算机学院
6. **课程的目的和任务**

《编译原理》是计算机专业的必修专业基础课，其他相关专业的血案秀课程，系统讲解编译程序设计相关的各种理论及技术，例如形式语言及自动机基础，词法分析器、语法分析器、语法制导翻译、存储分配机制等内容，具有一定的理论深度和学习难度。本课程的目标是使学生了解程序设计语言编译系统的基本理论、结构、方法及其功能；通过学习本课程使学生掌握设计和构造编译系统的一般原理、手段和基本实现技术。从而为今后进一步学习形式语言与自动机、软件工程形式化、形式逻辑、语义与语用学方法等理论以及进一步自学和提高打下一定的基础，特别为以后掌握各种编程语言进行软件开发、调试、测试、和实现提供了坚实的理论和实践基础。

1. **课程支撑的毕业要求指标点及课程目标**

**1. 本课程支撑的毕业要求指标点**

1. 指标点R1.3:具备计算机专业知识，并能够综合应用于解决计算机系统和计算机应用系统的设计与开发
2. 指标点R3.2:能够应用软件理论及专业知识，设计/实现软件模块/系统解决方案，体现创新意识
3. 指标点R10.1:具有有效的沟通与交流能力，能够清晰地表达思想、回应指令、撰写文稿

**2.本课程拟达到的教学目标**

通过本课程的理论学习与实践，使学生掌握阅读及编写程序设计语言编译程序代码的能力；理解计算机程序设计语言编译程序所涉及到的概念；明白编译系统的结构及功能；掌握设计和构造编译系统一般原理；进一步掌握应用具体某种高级程序设计语言编译程序的设计实现能力、手段和基本实现方法，培养学生的分析解决问题能力和设计实现软件的能力，为进一步开展相关领域的学习和科研打好良好基础。具体来说，主要教学目标包括：

1. 学生能够理解和应用基本的形式语言与自动机理论；
2. 学生能够阅读和设计编写模块级/系统的编译程序代码；
3. 学生能够准确、清晰地描述所开发程序的代码结构和功能。

**3.课程教学目标与毕业要求指标点的关系**

|  |  |  |  |
| --- | --- | --- | --- |
| 课程教学目标 | 毕业要求二级指标点 | | |
| R1.3 | R3.2 | R10.1 |
| 子目标1 | ✓ |  |  |
| 子目标2 |  | ✓ |  |
| 子目标3 |  |  | ✓ |

课程教学子目标1是培养学生能够理解和应用形式语言和自动机的基本概念和理论，具体包括：文法定义、文法分类、文法的推导、正规式、自动机定义、有限状态自动机的构造方法、文法和自动机的对应关系，这些内容是计算机专业知识及基本方法，可以应用于多种计算机应用系统的开发，因此能对毕业要求指标点R1.3的达成提供支撑。

课程教学子目标2是培养学生能够阅读和设计编写模块级/系统的编译程序代码，能够综合应用软件工程方法及编译程序基本知识，采取手工编写或者使用辅助工具实现指定高级程序设计语言的编译程序主体，包括词法分析模块、语法分析模块、语法制导翻译模块及目标代码生成模块。课程通过实验项目实践，设计报告讨论等环节来培养考核学生此方面的能力，因此可以为毕业要求指标点R3.2的达成提供支撑。

课程教学子目标3是培养学生能够准确、清晰地描述所开发程序的代码结构和功能，具体体现在学生撰写的实验报告应能够准确、清晰地描述实验项目的完成情况，包括实验项目的需求分析（功能描述）、程序设计、程序开发、代码测试等内容，因此可以对毕业要求指标点R10.1的达成提供支撑。

1. **课程教学方式/方法**

对于课程教学目标1，主要采用课堂教学的方式，通过知识教授、问题讨论等途径，使学生能够理解和应用形式语言与自动机的理论和方法。

对于课程教学目标2，主要采用设计分析作业和上机实验的方式，通过分模块实现的方式，使学生完成程序设计语言编译程序主体部分的设计及实现。

对于课程教学目标3，主要通过撰写实验报告的方式来训练学生，使之能够准确、清晰地描述所开发程序的代码结构和功能。

1. **课程教学内容与学时**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **一级知识点** | **二级知识点** | **三级知识点** | **掌握**  **层次** | **学时数**  **分配** |
| 课程介绍与基本概念 | 课程介绍 | 课程目标 | 记忆 | 2 |
| 课程内容概览 | 记忆 |
| 课程考核方式 | 记忆 |
| 编译程序的基本概念 | 编译程序的基本功能 | 记忆 |
| 翻译和解释 | 应用 |
| 编译程序基本结构 | 应用 |
| 形式语言与自动机基础 | 语言和文法 | 字母表和字符串 | 应用 | 4 |
| 语言 | 应用 |
| 文法及其形式定义 | 应用 |
| 文法的分类 | 应用 |
| 推导和短语 | 应用 |
| 分析树及二义性文法 | 应用 |
| 有限自动机 | 确定的有限自动机 | 应用 | 4 |
| 非确定的有限自动机 | 应用 |
| 有限自动机的确定化 | 应用 |
| 确定的有限自动机的化简 | 应用 |
| 有限自动机的应用 | 应用 |
| 正规表达式 | 正规表达式与自动机的等价性 | 应用 | 2 |
| 正规表达式与正规文法的等价性 | 应用 |
| 正规式、正规文法与确定的有限自动机的等价性 | 理解 |
| 词法分析 | 词法分析基础 | 词法分析程序与语法分析程序的关系 | 理解 | 2 |
| 词法分析程序的输入与输出 | 应用 |
| 记号的描述和识别 | 应用 |
| 词法分析程序的设计与实现 | 文法及状态转换图 | 应用 | 4 |
| 词法分析程序的构造 | 应用 |
| 词法分析程序的实现 | 应用 |
| 软件工具LEX/FLEX | 应用 |
| 4. 语法分析 | 语法分析简介 | 语法分析程序 | 理解 | 1 |
| 常用分析方法 | 理解 |
| 语法错误的处理 | 理解 |
| 自顶向下分析法 | 递归下降分析 | 应用 | 5 |
| 递归调用预测分析 | 应用 |
| 非递归调用预测分析基础 | 应用 |
| LL（1）分析 | 应用 |
| 自底向上分析方法 | 优先分析法 | 应用 | 2 |
| LR分析法 | 应用 | 4 |
| 软件工具YACC/BISION | 应用 | 2 |
| 语法制导翻译技术 | 语法制导翻译概述 | 语法制导定义 | 理解 | 4 |
| 语法翻译方案 | 理解 |
| 中间代码生成 | 中间代码形式 | 理解 |
| 自底向上的语法制导翻译 | 说明语句的翻译 | 应用 |
| 简单赋值语句的翻译 | 应用 |
| 布尔表达式的翻译 | 应用 |
| 控制语句的翻译 | 应用 |
| 过程调用语句的翻译 | 应用 |
| 语义分析 | 语义分析概述 | 语义分析的任务 | 理解 | 2 |
| 语义分析程序的位置 | 理解 |
| 语义分析阶段的错误处理 | 理解 |
| 符号表 | 符号表的建立和访问时机 | 理解 | 2 |
| 符号表内容 | 应用 |
| 符号表操作 | 应用 |
| 显式类型和静态类型检查 | 类型表达式 | 应用 | 2 |
| 类型等价 | 应用 |
| 类型检查其他主题 | 应用 |
| 目标代码生成 | 目标代码生成概述 | 目标代码生成程序的任务 | 理解 | 2 |
| 目标代码生成程序的位置 | 理解 |
|  | 目标代码生成程序设计的相关问题 | 理解 |
| 基本块与流图 | 基本块的划分方法 | 应用 | 1 |
| 流图 | 应用 |
| 下次引用信息 | 活跃变量 | 应用 | 1 |
| 下次引用计算算法 | 应用 |
| 目标语言及指令 | 目标机器描述 | 了解 | 2 |
| 代码生成算法 | 了解 |

**2.实验教学（16学时）**

（1）实验目的

编译原理设计实验教学的目的是为了让学生通过实验环境加深对编译程序开发中相关理论的认识，更好的掌握和设计编译程序结构和功能，体会软件开发过程；同时结合实际应用问题，让学生掌握相应编程方法和技巧。编译原理实验教学通过分阶段的四个实验，分别关注编译器设计的四个阶段：词法分析及语法分析、语义分析、中间代码生成及目标代码生成。每个实验均给出具体的实验要求及实验指导、测试样例。每个实验依赖于前面的实验，需按顺序进行，所有实验共同覆盖一个编译器的设计与实现全过程。

（2）实验内容及学时分配

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **实验名称** | **实验内容** | **实验软硬件环境要求** | **学时数分配** |
| 1 | 词法分析程序 | 1. 编写一个程序对使用类C语言书写的源代码进行词法分析，并打印分析结果。 2. 程序要能够检查源代码中可能包含的词法错误： 3. 最低要求1.1：能够识别词法中未定义的字符 4. 其他要求1.2:能识别指数形式的浮点数 5. 其他要求1.3:能识别8进制/16进制数   3）能定位错误位置  实验开展方式：  可以手工编写或者使用类似Flex工具编写 | 内存2GB以上/硬盘20GB以上的PC，安装：  1）GCC  2）Flex | 4 |
| 2 | 语法分析程序 | 1. 编写一个程序对使用类C语言书写的源代码进行语法分析，并打印分析结果。 2. 程序要能够检查源代码中可能包含的词法错误： 3. 最低要求2.1:能够输出语法分析后的树结点信息 4. 其他要求2.2:能识别常见的语法错误并给出具体位置 5. 其他要求2.3:能进行源代码中的错误恢复   3）能定位错误位置  实验开展方式：  可以手工编写或者使用类似Bison/Yacc工具编写 | 内存2GB以上/硬盘20GB以上的PC，安装：  1）GCC  2）Bison/Yacc | 4 |
| 3 | 语义分析程序 | 1. 在词法及语法分析程序的基础上，编写一个程序对使用类C语言书写的源代码进行语义分析及类型检查，并打印分析结果。 2. 程序要能够检查源代码中是否符合以下语义要求： 3. 最低要求3.1：能够实现对整型（int）和浮点型（float）变量的类型检查，两类变量不能相互赋值及运算；仅整型及浮点型变量才能参与算术运算 4. 其他要求3.2:能判断源代码是否符合以下语义假设并给出相应错误具体位置：函数仅能定义一次、程序中所有变量均不能重名、函数不可嵌套定义 5. 其他要求3.3:能检查结构体中域是否与变量重名，不同结构体中域是否重名   3）能定位错误位置  实验开展方式：  手工编写或者使用类似Bison/Yacc工具编写 | 内存2GB以上/硬盘20GB以上的PC，安装：  1）GCC  2）Bison/Yacc  。 | 4 |
| 4 | 目标代码生成程序 | 1. 在词法分析、语法分析及语义分析的接触上，编写一个程序对使用类C语言书写的源代码翻译为中间代码/目标代码，并打印翻译结果。 2. 程序要能够检查源代码中可能包含的词法错误： 3. 最低要求4.1:能够输出抽象语法树/四元式的中间代码，应至少包括以下代码类型：赋值语句、算术运算操作（加减乘除）、跳转语句、分支与循环语句 4. 其他要求4.2:能够将中间代码的基础上，生成某类目标代码（汇编/MIPS）   实验开展方式：  手工编写，在语义分析程序基础上添加代码生成部分 | 内存2GB以上/硬盘20GB以上的PC，安装：  1）GCC  2）Bison/Yacc | 4 |

（3）实验要求

a）学生不超过3人一组；

b）学生课前自学掌握C语言；

c）学生实验前在个人电脑上预装好gcc，flex，bison开发环境；

d）学生完成程序开发后，需要撰写实验报告。

（4）实验教材（讲义、指导书）

同课程教材。

（5）实验对课程教学目标的支持

实验题目均是以实际应用问题为背景。学生需要综合运用所学的C语言编程知识、软件工程、数据结构及算法等专业基础知识，使用相应工具（gcc/flex/bison）进行编译程序各子程序的设计与开发，并撰写实验报告准确清晰的描述软件结构和功能。以上学习实践过程可以支撑课程教学目标1、2、3。

1. **能承担此课的教师**

张敏、刘骥、葛亮、夏云霓、吴全旺

教学大纲制订者：张敏

教学大纲审定者：朱征宇