# 《编译技术》课程教学大纲

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程编号 | | | SENG 3157.02 | | | | 课程名称 | | | | | 编译技术 | | | | | |
| 课程英文名称 | Principles of Compiler | | | | | | | | | | | | | | | | |
| 总学时数 | 48 | 理论  学时 | | 32 | 实验  学时 | | | 0 | 上机  学时 | 16 | | | 本课程负责人 | | | 李贵林 | |
| 学 分 | 2 |
| 开课单位 | 软件学院 | | | | | | | 适用专业 | | 软件工程 | | | | | | | |
| 考核方式 | 期末笔试，作业 | | | | | | | | | | | | | | | | |
| 先修课程 | C语言程序设计、离散数学、数据结构 | | | | | | | | | | | | | | | | |
| 课程类型 | 专业选修课程 | | | | | | | | | | | | | | | | |
| 选用教材 | 《Compilers: Principles, Techniques and Tools》人民邮电出版社 [Sethi, Ullman Aho](http://book.douban.com/search/Sethi,%20Ullman%20Aho) | | | | | | | | | | | | | | | | |
| 主要教学  参考书 | 《编译原理》清华大学出版社 [张素琴](http://book.douban.com/search/%E5%BC%A0%E7%B4%A0%E7%90%B4) | | | | | | | | | | | | | | | | |
| 课程简介  （300-500字） | **课程性质：**编译原理是软件工程专业的一门重要的专业课程。通过本课程的学习，学生应掌握编译程序的结构及各部分的功能，掌握设计和构造程序语言编译系统的基本原理和技术。主要包括形式语言基础知识、词法分析器原理和构造方法、语法分析技术、语法制导翻译、运行时刻环境、符号表、中间代码生成、代码优化等基本方法和主要实现技术。  **课程主要内容包括：**编译器的构成、主要的词法分析算法与有穷自动机、自下而上语法分析（LL(1)分析、递归下降法等）、自上而下语法分析（LR分析法、算符优先分析法等）、语法制导翻译、运行阶段的存储组织与管理、符号表、代码优化与生成。  **课程目标1：**通过本课程的学习,学生应掌握高级程序设计语言的编译原理和基本技术、具体设计实现编译程序的基本技能，特别对形式语言和自动机、程序自动生成、语法制导翻译、数据流分析、控制结构分析及优化技术等基本原理和技术等基本原理和技术有较深入的了解。  **课程目标2：**了解编译器的实现方法，在此基础上培养起学生编写编译器的能力。 | | | | | | | | | | | | | | | | |
| 课程目标与毕业要求之间的关系 | 详细写明课程目标对应的毕业要求。  支持“**指标点1.3**”：针对软件工程及相关领域的复杂工程问题，设计软件系统的解决方案的能力。课程目标1主要实现指标点1.3中设计软件系统的能力；课程目标2主要实现指标点1.3中给出解决方案的能力。编译器作为计算机系统中最复杂的软件之一，本课程通过对该软件的设计思想和实现方法的讲解，以及动手编写必要的程序可以有效的培养学生在面对复杂工程问题时设计软件系统解决方案的能力。编译器概述从全局角度讲解编译器的设计思想。词法分析算法与有穷自动机、自下而上语法分析（LL(1)分析、递归下降法等）、自上而下语法分析（LR分析法、算符优先分析法等）、语法制导翻译、运行阶段的存储组织与管理、符号表、代码优化与生成各章分别讲解编译器各个环节的设计和实现过程。在每一章的学习中，都要求学生根据已学到的知识设计并编写自己的编译器的对应部分，从而锻炼学生提出解决方案的能力。本课程要求学生通过理论学习，课后作业等形式掌握编译器设计的思想方法，并通过阅读代码、设计模块和编程实现等方式将这些专业知识用于设计软件系统的解决方案。  **对应课程教学内容：**词法分析与有穷自动机：对词法分析器的要求、词法分析器的设计、正规表达式与有限自动机、词法分析器的自动生成；自下而上语法分析：LL（1）分析法、 递归下降分析程序的构造、预测分析程序、LL（1）分析中的错误处理；自上而下语法分析：算符优先分析、LR分析法；语法制导翻译：属性方法、基于属性方法的处理方法、S-属性文法的自下而上计算、L-属性文法的自顶向下翻译、自下而上计算继承属性；运行阶段的存储组织与管理：目标程序运行时的活动、运行进存储器的划分、静态存储分配、简单的栈式存储分配、嵌套过程语言的栈式实现、堆式动态存储分配；符号表：符号表的组织与作用、整理与查找、名字的作用范围、符号表的内容；代码优化与生成：局部优化、循环优化。 | | | | | | | | | | | | | | | | |
| 教学内容、方法、学时分配及所占期末考核比例 | **教学内容** | | | | | **学时** | | | | | **教学方法** | | | **题目类型** | | | **所占比例** |
| **授课** | | | **实验** | |  | | | |  | |  |
| 编译概述  了解和掌握高级程序设计语言与编译程序的关系；了解和掌握编译程序的功能 了解和掌握编译程序的体系结构；了解和掌握编译程序的工作过程；了解和掌握编译程序的组织方式；了解和掌握编译程序的构造方法。 | | | | | 2 | | | 0 | | 讲授法，实验教学，完成作业 | | | | 选择题，问答题，设计题 | | 5% |
| 词法分析与有穷自动机  本章主要介绍有穷自动机的理论及借助有穷自动机实现词法分析程序的方法。着重需要掌握以下内容：确定的有穷自动机DFA；非确定的有穷自动机NFA；NFA到DFA的转换；DFA的最小化；正规表达式到DFA的转换；单词的形式；词法分析程序的设计方法 | | | | | 4 | | | 2 | | 讲授法，实验教学，完成作业 | | | | 选择题，问答题，设计题 | | 10% |
| 自上而下语法分析  自上而下语法分析的基本思想；自上而下语法分析面临的问题及解决方法；消除左递归的方法；FIRST（x）的求法；FOLLOW（U）的求法；递归子程序的构造方法；LL（1）文法；LL（1）分析表的构造方法；自上而下语法分析时为避免回溯对文法的要求。 | | | | | 4 | | | 4 | | 讲授法，实验教学，完成作业 | | | | 选择题，问答题，设计题 | | 20% |
| 自下而上语法分析  掌握自下而上语法分析的基本思想；自下而上语法分析面临的问题及解决方法；简单优先分析法；算符优先分析法(算符优先文法、算符优先关系及其构造)；优先函数的构造；LR分析器；活前缀；LR(0)项目、以LR(0)项目集为状态的识别活前缀的DFA；LR(1)项目、以LR(1)项目集为状态的识别活前缀的DFA；LR(0)分析表的构造方法；SLR(1)分析表的构造方法；LR(1)分析表的构造方法；LALR(1)分析表的构造方法；LR(0)文法、SLR(1)文法、LR(1)文法、LALR(1)文法；二义性文法的应用。 | | | | | 8 | | | 4 | | 讲授法，实验教学，完成作业 | | | | 选择题，问答题，设计题 | | 30% |
| 语法制导翻译  掌握语义分析方法中常见的语法制导翻译基本思想；常见的中间语言形式：逆波兰表示、四元式表示、三元式表示；常见语法成分的逆波兰表示、四元式表示、三元式表示、三地址代码表示；程序设计语言中各种常见语法成分的目标代码结构；常见语法成分的属性文法或翻译方案。 | | | | | 4 | | | 2 | | 讲授法，实验教学，完成作业 | | | | 选择题，问答题，设计题 | | 15% |
| 运行阶段的存储组织与管理  掌握运行阶段的存储组织与分配的基本思想；静态存储分配的基本思想；动态存储分配的基本思想；活动记录；以过程为单位的栈式动态存储组织与分配的基本思想；嵌套结构语言非局部量访问的实现；不同形实参数结合方式。 | | | | | 4 | | | 2 | | 讲授法，实验教学，完成作业 | | | | 选择题，问答题，设计题 | | 10% |
| 符号表、代码优化与生成  掌握符号表的作用；符号表中的内容；嵌套结构语言的栈式符号表。代码优化的基本思想；什么是基本块；如何划分基本块；如何构造程序的流图；如何确定结点的必经结点集；什么是回边；如何确定循环；怎样利用基本块的DAG图进行基本块的优化；如何进行循环的优化。 | | | | | 4 | | | 2 | | 讲授法，实验教学，完成作业 | | | | 选择题，问答题，设计题 | | 10% |
| 期末知识点复习。 | | | | | 2 | | | 0 | | 讲授法 | | | |  | |  |
| 课程的评价与持续改进机制 | 1. 考核采取平时作业和期末笔试综合评定成绩。其中，平时作业成绩占总成绩30%，期末笔试占总成绩70%。 2. 在教学准备和内容上，积极进行课程改革研究，跟踪课程内容的变化，关注本课程相关知识的最新发展动态，适当地将相关前沿知识引入到课程教学中。 3. 在教学方法和手段上： 4. 明确本门课程与培养目标和毕业要求之间的关系，激发学生的内在动力，提高学习积极性。 5. 教学方法上加强教学互动，设计更合理的提问或讨论方式与学生进行交流沟通； 6. 教学内容注重知识点在实际中的应用。在教学中，应注重引入相关案例，使学生了解所学将抽象的概念与在编程中遇到的实际案例联系起来，加深学生对所学知识的理解，然后讲授分析具体的各个知识点的相关内容与用法。 7. 根据督导组反馈和学生评教的意见和建议，对本年度的教学环节予以改进。 | | | | | | | | | | | | | | | | |