

# 中国科学技术大学本科新开课程申请表

2012年4月4日

课程名称	编译原理和技术 (H)			
英文名称	Principles and Techniques of Compilers (H)			
开课教师	姓名	职称	年龄	所在院系
				计算机科学技术学院
面向对象	计算机科学技术学院英才班本科生			
预修课程	数据结构、汇编语言程序设计、C 程序设计语言			
课程属性	<input type="checkbox"/> 通修 <input type="checkbox"/> 学科群基础 <input checked="" type="checkbox"/> 专业 <input type="checkbox"/> 高级		<input type="checkbox"/> 选修 <input checked="" type="checkbox"/> 必修	<input type="checkbox"/> 春 <input type="checkbox"/> 夏 <input checked="" type="checkbox"/> 秋 <input type="checkbox"/> 冬
	理论学时/实验学时		60/40	学分
				4
		<i>理论课 20 学时 1 学分, 实验 40 学时 1 学分</i>		新编课号
计划调整	<input type="checkbox"/> 原计划中的必修学分不变。 <input checked="" type="checkbox"/> 原计划中的必修学分从 <u>3.5</u> 变为 <u>4</u> 。			
	<input type="checkbox"/> 替换原计划中的课程：			
	课程编号	课程名称		学分/学时
	01113300	编译原理和技术		3.5/60+30
申请原因：				
学院意见：				
教务处意见：				
备注：				

## 理论教学大纲[2019年8月修订]

大纲撰写人	张昱
一、教学目标和基本要求	
<p>本课程是高校计算机科学与技术专业的专业基础课。目的是让学生对程序设计语言的设计和实现技术有深刻的理解，对和程序设计语言有关的理论有所了解，并能把本课程讨论的概念和技术用到软件设计和开发、新程序设计语言和新计算机体系结构的设计和实现中。</p>	
二、课程简介	
<p>本课程介绍编译器构造的一般原理、基本设计方法和主要实现技术，其内容包括词法分析、语法分析、语法制导的翻译、语义分析、运行时存储空间的组织和管理、中间表示与中间代码生成、目标代码生成、代码优化、编译系统和运行系统等。除了介绍命令式编程语言的编译技术外，本课程还选择性函数式语言和介绍面向对象语言的实现技术、数据流分析、过程间分析、内存管理与垃圾收集、循环优化以及并行编译等高级内容。本课程还强调一些相关的理论知识，如形式语言和自动机理论、语法制导的定义和属性文法、类型论和类型系统、数据流分析的一般框架等。</p>	
三、教学重点、难点	
<p>教学重点：语法分析的各种方法、语法制导的翻译、运行时存储空间的组织和管理、中间表示及代码生成。</p> <p>教学难点：LR 与 LL(*)分析、数据流分析的一般框架。</p>	
四、教材名称及主要参考书	
<p>教材：陈意云、张昱.《编译原理》(第三版), 北京: 高等教育出版社, 2014。</p> <p>主要参考书:</p> <p>[1] Alfred V.Aho Monica S.Lam Ravi Sethi Jeffrey D.Ullman . <i>Compilers: Principles, Techniques and Tools</i>. 2nd, ed. New York: Addison-Wesley, 2007.</p> <p>[2] Andrew W.Appel, Maia Ginsburg. <i>Modern Compiler Implementation in C</i>. Cambridge University Press, 2004.</p> <p>[3] Andrew W.Appel, Jens Palsberg. <i>Modern Compiler Implementation in Java</i>. 2nd, ed. Cambridge University Press, 2002.</p> <p>[4] Steven S.Muchnick. <i>Advanced Compiler Design and Implementation</i>. Hong Kong: Morgan Kaufmann Publishers, 1997.</p> <p>[5] John R. Levine. <i>Linkers and Loaders</i>. Hong Kong: Morgan Kaufmann Publishers, 1999.</p>	
五、课程章节主要内容及学时分配	
<b>第1章 引论 (1.5 学时)</b> 通过简要描述编译器的各个组成部分以及编译器技术的各种应用来介绍编译这个课题。 1.1 编译器概述 (1 学时) 在传统编译基础之上, 增加对交叉编译、增量编译、即时编译、预先编译的介绍。 1.2 编译器技术的应用 (0.5 学时)	

## 第 2 章 词法分析 (4.5 学时)

重点围绕词法分析器的自动生成展开, 先介绍与之有关的正规式和有限自动机概念, 然后介绍词法分析器的自动生成方法, 最后介绍一个词法分析器自动生成工具 Lex。

- 2.1 词法记号及属性 (0.5 学时)
- 2.2 词法记号的描述与识别 (1 学时)
- 2.3 有限自动机 (2 学时)
- 2.4 从正规式到有限自动机 (0.5 学时)
- 2.5 词法分析器的生成器 (0.5 学时)

## 第 3 章 语法分析 (8 学时)

阐述编译器采用的典型语法分析方法。首先提出有关上下文无关文法的基本概念, 然后介绍适合于手工实现的预测分析技术, 最后给出语法分析器自动工具使用的 LR、LL(\*)分析算法。本章还扩展所介绍的分析方法, 使之能从常见的错误中恢复过来, 增加对业界使用较多的分析器生成工具 ANTLR 的介绍。

本次修订~~删减~~对不同 LR 分析技术的深入介绍, 例如不深入介绍 LR、LALR 的分析策略及其文法书写和分析的对应关系, 仅详细介绍移进-归约以及 SLR 分析, 说明 LR 与 LL、LL(\*)的本质区别, 各自的优劣。

- 3.1 上下文无关文法 (1 学时)
- 3.2 语言和文法 (1.5 学时)
- 3.3 自上而下分析 (1.5 学时)
- 3.4 自下而上分析 (0.5 学时)
- 3.5 LR 分析器 (~~3 学时~~ **删减, 1 学时**)
- 3.6 LL(\*)分析 (**新增, 1 学时**)
- 3.7 二义文法的应用 (1 学时)
- 3.8 语法分析器的生成器 (0.5 学时)

## 第 4 章 语法制导的翻译 (4 学时)

讨论语义规则和文法的产生式相联系的两种方式: 语法制导的定义和语法制导的翻译方案, 还讨论它们的实现方法。

本次修订~~删减~~属性的自下而上计算, 而**补充**结合 LL(\*)分析的自上而下计算。

- 4.1 语法制导的定义 (1 学时)
- 4.2 S 属性定义的计算 (1.5 学时)
- 4.3 L 属性定义的计算 (1.5 学时)

## 第 5 章 类型检查 (5 学时)

以类型检查为代表, 介绍编译器的静态检查工作。主要介绍简单类型系统的设计、多态类型概念、类型表达式的等价和算符的重载。

- 5.1 类型在编程语言中的作用 (1 学时)
- 5.2 描述类型系统的语言 (0.5 学时)
- 5.3 一个简单类型检查器的规范 (1 学时)
- 5.4 多态函数 (1 学时)
- 5.5 类型表达式的等价 (0.5 学时)
- 5.6 函数和算符的重载 (1 学时)

## 第 6 章 运行时存储空间的组织和管理 (7 学时)

在考虑代码生成之前，需要讨论静态的程序正文和运行时的过程活动之间的联系，考察静态的名字和运行时数据对象之间的绑定关系。不仅要讨论一个活动记录中的数据安排，还要讨论程序执行过程中，所有存活的过程活动的活动记录的组织方式。此外，还需要考虑运行时的动态内存管理途径与技术、即时编译与动态重新编译、虚拟机的一些通用特征等。

本次修订**增加**被广泛应用于许多现代编程语言的函数闭包及其实现，以及垃圾收集、即时编译与动态重新编译等内容。

6.1 局部存储分配 (1 学时)

6.2 全局栈式存储分配 (2 学时)

6.3 非局部名字的访问 (1 学时)

6.4 参数传递 (0.5 学时)

6.5 高阶函数及闭包 (新增, 0.5 学时)

6.6 动态内存管理途径与技术 (新增, 1 学时)

6.7 其他现代编译系统的运行时组织与管理 (新增, 1 学时)

## 第 7 章 中间表示与中间代码生成 (6 学时)

介绍几种常用的中间表示，包括现代编译框架所使用的中间表示及其特征；然后介绍用语法制导定义或遍历抽象语法树的方法描述把编程语言的构造翻译成独立于机器的中间表示，后一种方法常用于现代编译器，故作为重点来讲解。

本次修订**调整**中间代码生成所基于的方法，由边分析边生成的语法制导定义**调整**为通过遍历分析产生的抽象语法树来进行中间代码生成；**增加**现代编程框架，特别是深度学习框架，所引入的中间表示新技术。

7.1 中间语言 (1 学时)

7.2 声明语句 (1 学时)

7.3 赋值语句 (1 学时)

7.4 布尔表达式和控制流语句 (1 学时)

7.5 现代编程框架中的中间表示 (新增, 2 学时)

## 第 8 章 代码生成 (4 学时)

基于一种简单的机器模型，介绍一种简单的代码生成算法。

本次修订将**增加**现代异构体系下的代码生成技术的高级话题。

8.1 代码生成器设计中的问题 (0.5 学时)

8.2 目标语言 (0.5 学时)

8.3 基本块和流图 (1 学时)

8.4 一个简单的代码生成器 (1 学时)

8.5 异构体系下的代码生成 (1 学时)

## 第 9 章 独立于机器的优化 (10 学时)

先通过一个简单的实例来了解代码优化；然后介绍数据流分析，它包括几类重要的全局收集且随后用于代码改进变换的信息。进一步介绍数据流分析框架的一般思想、程序中循环的发现和析、过程间分析，最后基于这些基本方法选择性介绍常量传播、部分冗余删除优化、并行性分析等。

以往对本章的讲授偏少，本次修订将增强这部分内容的具体实施。

9.1 优化的主要种类 (2 学时)

9.2 数据流分析介绍 (2 学时)





## 附注说明：

- 1、理论课请附上教学大纲；
- 2、如课程中含有有实验，请同时附上实验教学大纲；
- 3、如是实验课，只需提交实验教学大纲；
- 4、请一定认真填写计划调整一栏。

各专业若需要向培养计划中添加新课程，都须提交本科新开课程申请表，经院系和教务处同意后，加入培养计划。

院系及教务处在审核新开课程时，须遵循教学培养计划制定规范，审核加入新开课程计划的总学分及学分分配比例，认真审阅申请表“计划调整”一栏。

此申请表在提交专业培养方案、修改教学信息和修改教学任务过程中需添加新课时使用。