

CERACU

全国高等学校计算机教育研究会团体标准

关于编译课程规范的一些思考

Specification for Core Courses of Computer Science

课程标准工作汇报及问题探讨 ←

武汉大学 杜卓敏

计算机核心 课程标准

CERACU

2019首批课题
委托牵头单位

- ▶ 操作系统
北京大学 陈向群
- ▶ 程序设计
清华大学 郑莉
- ▶ 数据结构
重庆大学 王茜
- ▶ 编译原理
武汉大学 何炎祥
- ▶ 软件工程
国防科技大学 毛新军



目录

CONTENTS

1

课程标准研究进展

2

课程标准框架

3

课程标准正文

4

课程标准附录案例

5

后续工作

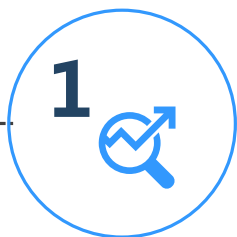
1

课程标准研究进展



标准研究时间表

Compiler Principles



2018年11月

成立总体组

熟悉相关资料

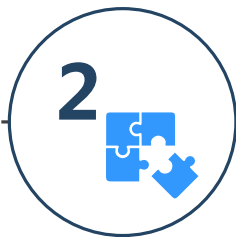
建议标准的目的和范围

撰写标准研制方案

2018年12月

组成5个标准工作组

确定标准的目的和范围



2019年1-3月

确定标准总体框架



2019年4-11月

完成标准征求意见稿

2019年12月

第一轮征求意见



2020年1-3月

按照第一轮征求意见结果
修改标准文本

撰写标准应用指南/案例

2020年4月-5月

第二轮征求意见和修改，
形成报批稿

2

课程标准框架



计算机核心课程标准研讨

RESEARCH METHODS AND IDEAS



计算机核心课程标准研讨会 2019.6.15 清华大学

工作进展情况汇报
标准草案通读研讨
共性问题讨论
确定下一步研究计划

确定五门课程标准的正文格式和框架必须统一。

标准中仅给出课程的知识体系和教学要求的规范定义，而不规定课程具体教学要求，将课程设计的自由度留给学校。

通过资料性附录给出若干典型的课程规范，便于参考实施。

课程标准的范围和内容框架

Compiler Principles

范围

本标准给出了高等学校编译原理课程的知识点集合、知识体系结构，规定了课程的教学要求。

本标准适用于高等学校编译原理课程、教材、教学过程和教学评价。

内容框架

知识体系按照分层定义，结构层级不超过三级，每个知识点可以容纳下级知识点，也可以是原子知识点。

能力要求给出了解、掌握、综合应用三个层次的具体要求。

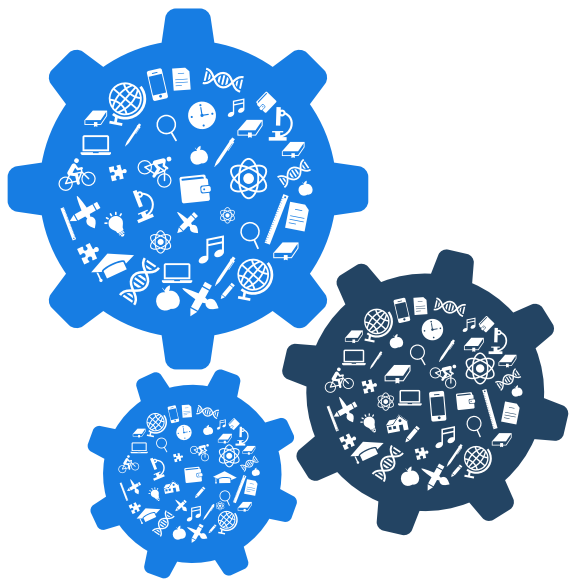
学时要求是应至少达到的学时下限，不设上限，包括总学时、理论学习学时（包括听课、自学的各种形式）、实践学时（包括实验课、项目实训等各种形式），“其他”可根据知识点的需要扩展。

课程开设单位遵循本标准

从本标准中选择适用的知识点和能力要求、学时要求，组成一门具体课程的内容和要求。

考核知识点应覆盖所有被选择作为该课程内容的知识点，考题难度应与该课程所选的各知识点能力要求、学时要求相符。每道题目应明确对应到一个或多个知识点及能力要求。各知识点题目比例应与学时比例一致，含多个知识点的综合题目直接按学习次序靠后的知识点和难度较大的一个或多个知识点计。

考核形式应适合知识点的内容和能力要求。



3

课程标准正文



知识点层级

Specification for Core Courses of Computer Science
Compiler Principles

编译课程的知识体系



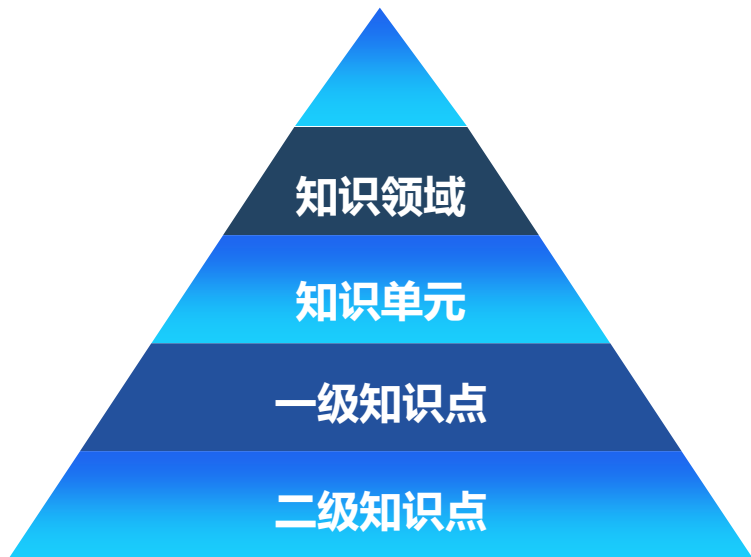
- 标准主要包括编译原理课程的基础知识点及要求，但不限于这些基础知识点。所列基本知识点，课程授课和学习中应该以多种教学方式充分覆盖。应包括本标准规定的知识点80%以上。
- 主讲教师可根据学生特点和不同培养目标的要求，采用不限于课内教学的方式引导和教授其它扩展知识点，扩大学生知识面。
- 本标准中未包含的知识点应作为扩展知识点，按照本标准规定的格式进行描述。

知识点层次

根据专业知识体系的要求，知识体系分为若干知识领域，知识领域下设若干知识单元，知识单元由若干知识点组成。课程中主要涉及知识单元与知识点。

编译课程的知识单元

知识领域	知识单元（核心）	知识单元（一般）
CS-CP编译原理	CP01. 语言及其处理 CP02. 语言及其描述 CP03. 词法分析 CP04. 语法分析 CP05. 语义分析及中间代码生成	CP06. 运行环境 CP07. 代码优化 CP08. 目标代码生成



知识单元类型

核心知识单元：重点教授

要求学生理解并熟练掌握

一般知识单元：要求学生了解

可以选讲或者布置学生自学

编译课程的知识单元和知识点（部分）

知识单元	标识符	知识点名称	知识点内容	类型	备注
CP01. 语言及其处理	CP01_01	语言	程序设计语言，低级语言，机器语言，汇编语言，高级语言等。	1	
	CP01_02	语言的处理器	语言及其处理，语言的处理器，语言的翻译，编译技术的应用领域等。	1	
	CP01_03	程序的翻译	翻译方式，编译方式，解释方式，翻译程序，编译程序/器，解释程序/器，汇编程序/器等。	2	
	CP01_04	编译方式和解释方式	目标代码，编译程序，汇编程序，解释程序，调试，编译方式和解释方式的区别、特点和适用范围等。	1	
	CP01_05	编译程序的工作过程	分析，综合，词法分析，语法分析，语义分析，中间代码生成，代码优化，目标代码生成，符号表管理，出错处理等。	2	
	CP01_06	编译程序的结构	编译程序的逻辑结构等。	1	
	CP01_07	编译程序的组织方式	遍/趟/趟程，单趟扫描，多趟扫描，前端，后端等。	1	
	CP01_08	编译程序的构造	高级语言的自编译性，自展，编译程序的移植，交叉编译器，编译程序的自动生成等。	1	选讲或自学

旧版本草案
对知识点重要性进行了分类

知识点类型

- 1一般知识点 要求学生了解，限于课时，也可以选讲或者布置学生自学。
- 2重要知识点 核心知识点（重要知识点和难度知识点）需要重点教授，并要求学生理解并熟练掌握
- 3难度知识点

不需要列出全部课程相关知识点，仅列出需要讲授和需要学生自我学习的知识点。备注中注明了可采用选讲或者自学方式的知识点。



知识单元	一级知识点		二级知识点		备注
	标识符	名称	标识符	名称	
CP04. 语法分析	CP04_01	语法分析			
	CP04_02	语法规则的形式化表示			
	CP04_03	语法分析程序			
	CP04_04	自顶向下语法分析	CP04_04_01	最左推导	
			CP04_04_02	带回溯的自上而下分析方法	
			CP04_04_03	确定的自上而下分析方法	
			CP04_04_04	消除左递归	
			CP04_04_05	避免回溯	
	CP04_05	语法分析模型	CP04_05_01	自顶向下语法分析器	
			CP04_05_02	自底向上语法分析器	
	CP04_06	LL分析方法	CP04_06_01	匹配-推导语法分析技术	
			CP04_06_02	SELECT集	
			CP04_06_03	FIRST集	
			CP04_06_04	FOLLOW集	
		CP04_06_05	LL(1)文法		
		CP04_06_06	LL分析器		
		CP04_06_07	LL分析算法		
		CP04_06_08	LL(1)分析表		
		CP04_06_09	二义性文法的LL分析		
CP04_07	递归子程序法				

知识点层级举例

Compiler Principles

编译课程的知识单元和知识点 (部分)

知识单元	一级知识点		二级知识点		备注	
	标识符	名称	标识符	名称		
CP04. 语法分析	CP04_08	自底向上语法分析	CP04_08_01	句柄		
			CP04_08_02	最左归约和句柄剪枝		
			CP04_08_03	移进-归约语法分析技术		
	CP04_09	优先分析方法	CP04_09_01	简单优先分析方法		选讲或自学
			CP04_09_02	算符优先分析方法		
	CP04_10	LR分析方法	CP04_10_01	LR(k)文法		
			CP04_10_02	LR分析器		
			CP04_10_03	LR分析算法		
			CP04_10_04	LR分析表		
			CP04_10_05	规范句型活前缀		
			CP04_10_06	LR(0)项目		
			CP04_10_07	LR(0)项目集规范族		
			CP04_10_08	识别规范句型活前缀的DFA		
			CP04_10_09	LR(0)分析方法		
			CP04_10_10	SLR(1)分析方法		
CP04_10_11			LR(1)项目			
CP04_10_12			LR(1)项目集规范族			
CP04_10_13	LR(1)分析方法					
CP04_10_14	LALR(1)分析方法					
CP04_10_15	二义性文法的LR分析					
CP04_11	语法分析器自动生成					
CP04_12	语法错误和错误恢复					



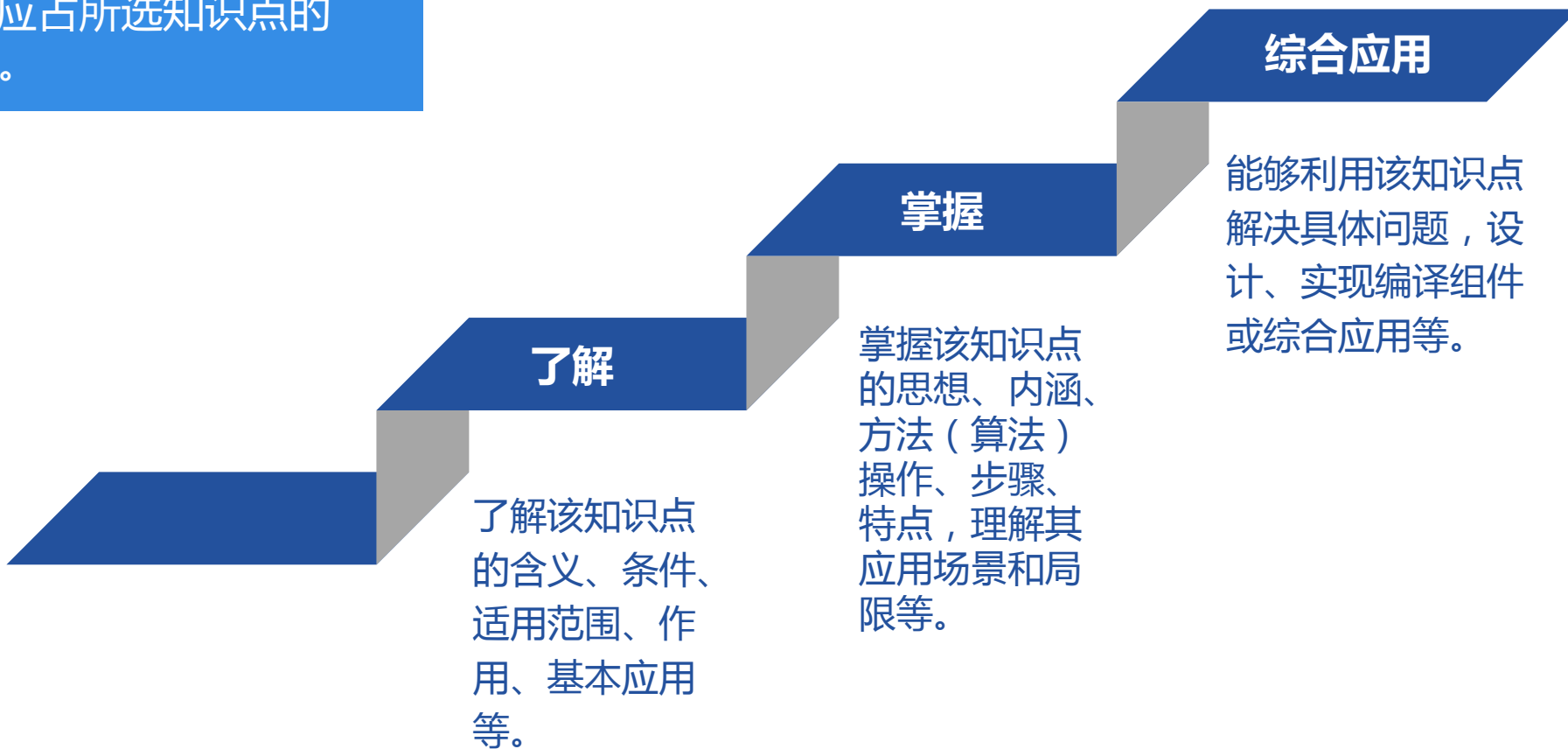
能力要求层级

Specification for Core Courses of Computer Science
Compiler Principles

02 能力要求层级

Compiler Principles

能力要求为“掌握”、“综合应用”的应占所选知识点的80%以上。



3

2

1



知识点及要求

Specification for Core Courses of Computer Science
Compiler Principles

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04	语法分析 1	语法，语法分析	正确理解语言、语言的语法；了解自顶向下和自底向上语法分析的主要思想；理解语法分析器的工作步骤。	掌握自顶向下和自底向上语法分析方法的主要思想、特点、分析器结构、分析过程、分析算法、异同和适用范围。	能评价各种分析方法的优缺点，并根据文法和语言的特点，选择合适的语法分析技术和语法分析的求解模型（下推自动机），采用相应的语法分析算法和分析器结构完成语法分析工作。	32	16	16	
CP04_01	语法分析	语法，语法规则，语法规则语法错误等。	了解语法分析的任务、作用和语法分析器的工作原理。	掌握语法分析的基本思想；理解语法分析器的工作原理。	能够选择合适的语法分析技术和语法分析的求解模型（下推自动机），采用相应的语法分析算法和分析器结构完成语法分析工作。	0.5	0.5	0	
CP04_02	语法规则的上下文无关文法形式化表示法 CFG，下推自动机 PDA，BNF 描述）。	能够理解程序设计的文法（如上下文无关文法）；理解下推自动机 PDA 的结构。	掌握语言的语法规则的形式化表示（上下文无关文法）；理解下推自动机 PDA 的结构。	能对基本的语法结构进行形式化表示；能够正确描述采用下推自动机进行语法分析的过程。	3.5	0.5	3		
CP04_03	语法分析程序	语法分析程序/语法分析器/分析器的结构语法分析树，抽象语法树 AST 等。	了解语法分析程序的工作原理和逻辑结构。	理解语法分析器的工作原理、逻辑结构和分析算法。	熟悉语法分析器的工作原理和逻辑结构；能够采用合适的语法分析算法，设计和实现语法分析程序。	0.5	0.5	0	

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_04	自顶向下语法分析	自顶向下语法分析/自上而下分析的主要思想，语法分析的主存在的问题和解决方案，方法、限制和局限等	了解自顶向下语法分析/自上而下分析的主要思想，语法分析的主存在的问题和解决方案，方法、限制和局限等	理解自顶向下语法分析方法及其限制和局限等；熟悉自顶向下语法分析器的工作原理、逻辑结构和分析算法；理解和掌握LL(1)分析方法和（或）递归子程序法；熟悉预测分析器的总体结构和预测分析表的手工构造。	熟练掌握自顶向下语法分析技术；熟悉语法分析器的工作原理、逻辑结构和分析算法；理解LL(1)分析方法和递归子程序法；能够选择合适的自顶向下语法分析技术，对程序设计语言的基本语法成分进行语法分析，手工构造语法分析器，并对其优势和局限性进行评价分析。	1	1	0	
CP04_04_01	最左推导	最左推导，最左直接推导，最左句型，自上而下生成语法树等。	理解最左推导，最左直接推导，最左句型，自上而下生成语法树等。	理解最左推导、自上而下生成语法树在自上而下语法分析中的运用。	能够运用最左推导、自上而下生成语法树来描述符号串的自上而下语法分析过程。				
CP04_04_02	带回溯的自上而下分析方法	不确定的自上而下分析方法作用和危害；理解带回溯，带回溯的自上而下危害，方法的局限等。	了解带回溯、带回溯的自上而下分析方法作用和危害；理解带回溯的自上而下危害，方法的语法分析过程。	理解不确定的自上而下分析方法，方法的局限等。	掌握不确定的自上而下分析方法；能够描述符号串的回溯的自上而下分析过程；并能够设计实现带回溯的自上而下语法分析器。				
CP04_04_03	确定的自上而下分析方法	确定的自上而下分析的条件左递归，回溯消除左递归，避免回溯等。	了解确定的自上而下语法分析的基本思想、面临的问题和解决方法；理解左递归、回溯。	理解确定的自上而下分析方法，方法的局限等。掌握左递归的消除和避免回溯的条件。	掌握确定的自上而下分析方法；能够描述符号串的不带回溯的自上而下分析过程；并能够设计实现确定的自上而下语法分析器。				

编译课程的知识点及要求 (部分)

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_04_04	消除左递归	消除左递归,了解左递归、直接消除直接左递归、间接左递归,EBNF,直归、EBNF等。接改写法,消除间接左递归等。	理解和掌握直接左递归的消除、间接左递归的消除方法。	能够运用左递归消除方法,消除文法中的左递归。					
CP04_04_05	避免回溯	左公因子,提取左公因子等件。	了解避免回溯的条件。	理解避免回溯的条件回溯的消除。	能够通过提取左公因子等方法对文法进行等价变换,消除回溯。				
CP04_05	语法分析模型	下推自动机PDA,下推栈PDA的结构,LL分析器总体结构,LR分析器总体结构等。	了解下推自动机PDA的结构,掌握语法分析器的总体结构和工作过程。	理解下推自动机PDA的结构,掌握语法分析器的总体结构和工作过程。	能够利用语法分析模型,构造自上而下/自下而上语法分析器。	0.5	0.5	0	
CP04_05_01	自顶向下语法分析器	自顶向下语法分析器,递归分析的基本思想;下降分析器,熟悉语法分析器的LL语法分析器工作原理、逻辑结构和分析算法等。	掌握自顶向下语法分析器,递归分析的基本思想;下降分析器,熟悉语法分析器的LL语法分析器工作原理、逻辑结构和分析算法等。	理解和掌握LL(1)分析方法和递归子程序法;熟悉预测分析器的总体结构和预测分析表的手工构造。	能够实现简单算术表达式的递归下降分析器。能够选择合适的自顶向下语法分析技术,对程序设计语言的基本语法成分进行语法分析,手工构造语法分析器,并对其优势和局限性进行评价分析。				
CP04_05_02	自底向上语法分析器	自底向上语法分析器,LR语法分析器等。	掌握自底向上语法分析器,LR语法分析的基本思想;熟悉语法分析器的工作原理、逻辑结构和分析算法等。	理解优先分析方法的主要思想及其局限;理解和掌握LR分析方法及其优势;熟悉LR分析器的总体结构和LR分析表的手工构造。	能够理解算符优先分析方法。能够选择合适的LR语法分析技术(LR(0),SLR(1),LR(1),LALR(1)),对程序设计语言的基本语法成分进行语法分析,手工构造语法分析器,并对其优势和局限性进行分析评价。				

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_06	LL分析方法	LL分析方法的了解LL分析方法的主要思想，LL分析的条件，LL(1)文法，LL(1)分析器，LL(1)分析表，LL(1)分析算法和过程，非递归预测分析，二义性文法的LL分析等。	理解LL分析方	理解和掌握LL分析方	能够采用LL分析方法进行非递归预测分析；能够构造文法的LL(1)分析表，设计和实现LL(1)分析器。能够通过冲突消解，实现二义性文法的LL分析。	6	2	4	
CP04_06_01	匹配-推导语法分析技术	匹配-推导语法分析技术，匹配-推导语法分析中的冲突等	了解匹配-推导语法分析技术的主要思想。	理解匹配-推导语法分析中的冲突；能够描述符号串的匹配-推导语法分析过程。	能够运用匹配-推导语法分析技术进行语法分析。				
CP04_06_02	SELECT集	SELECT集，了解SELECT集的意义，避免回溯的条件，SELECT集的求法等。	理解SELECT集的含义，意义，避免回溯的条件，掌握SELECT集的求法等。	能够利用SELECT集，构造LL(1)分析表；进行LL(1)文法的判定。					
CP04_06_03	FIRST集	FIRST集，了解FIRST集的意义，FIRST集的求法等。	理解FIRST集的含义、意义，掌握FIRST集的求法等。	能够熟练掌握FIRST集的求法；能够编程实现求FIRST集。					

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_06_04	FOLLOW集	FOLLOW集，了解FOLLOW集，FOLLOW集的意义，FOLLOW集的求法等。	了解FOLLOW集的意义，FOLLOW集的求法等。	理解FOLLOW集的含义、意义，掌握FOLLOW集的求法等。	能够熟练掌握FOLLOW集的求法；能够编程实现求FOLLOW集。				
CP04_06_05	LL(1)文法	LL(1)文法，LL(1)语言，LL(k)文法，LL(1)文法，二义性文法等。	了解LL(1)文法，LL(1)语言的定义、判定、应用等。	理解和掌握LL(1)文法、LL(1)语言的定义和判定；理解LL(k)文法，非LL(1)文法，二义性文法等。	能够熟练掌握LL(1)文法的判定；并能用LL(1)文法描述LL(1)语言。				
CP04_06_06	LL分析器	LL(1)分析器，分析表，符号栈，LL(1)分析程序等。	了解LL(1)分析器的结构和工作原理。	理解和掌握LL(1)分析器的结构和工作原理；理解分析表，符号栈分析，手工构造LL(1)语法分析器LL(1)分析程序等；能对其优势和局限性进行评价。	能采用LL(1)分析方法，对程序设计语言的基本语法成分进行语法分析。				
CP04_06_07	LL分析算法	非递归预测分析算法，LL(1)分析算法等。	了解非递归预测分析方法，LL(1)分析算法等。	理解和掌握非递归预测分析方法，LL(1)分析算法等。	能够设计和实现LL(1)分析算法。				
CP04_06_08	LL(1)分析表	LL(1)分析表，LL(1)分析表的构造，冲突等。	了解LL(1)分析表的构造。	掌握LL(1)分析表的构造，理解冲突，能用LL(1)分析表进行语法分析。	能够设计和实现构造LL(1)分析表的算法，编程实现LL(1)分析表的自动构造。				

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_06_09	二义性文法的LL分析	二义性文法，理解二义性文法；二义性文法的了解无二义性规则冲突消解，无二义性文法的LL分二义性规则，析。二义性文法的LL分析等。	理解二义性文法，二义性文法的冲突消解理解无二义性规则，二义性文法的LL分析。	能够通过二义性规则的使用，消解冲突，设计和实现二义性文法的LL分析。					
CP04_07	递归子程序法	递归子程序法/了解递归子程序法/递归下降分析递归下降分析方法的主要思想，递归下降分析的条件，递归下降分析算法，递归子程序的构造等	理解和掌握递归子程序法的主要思想、条件和分析算法，掌握递归子程序的构造等。	熟练掌握递归子程序的构造，能够编程设计和实现递归子程序。	1.5	0.5	1		
CP04_08	自底向上语法分析	自底向上语法分析/自下而上分析的主要思想，存在方案。的问题，解决问题的方案，方法的限制和局限等	了解自底向上语法分析方法及其限制和局限等；掌握语法分析器的工作原理、逻辑结构和分析算法；熟悉LR分析器的总体结构和LR分析表的手工构造。	熟练掌握自底向上语法分析技术；熟悉语法分析器的工作原理、逻辑结构和分析算法；理解优先分析方法的主要思想及其局限；理解和掌握LR分析方法及其优势；能够选择合适的自底向上语法分析技术，对程序语言的基本语法成分进行语法分析，手工构造语法分析器，并对其优势和局限进行评价分析。	1	1	0		

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_08_01	句柄	短语，简单了解短语，简单（直接）短语（直接）短语，句柄，素短语柄等。最左素短语，句柄的条件等	理解短语，简单（直接）短语，句柄，以及它们和语法分析树的关系；了解素短语最左素短语；理解句柄的条件等。	理解短语，简单（直接）短语，句柄，以及它们和语法分析树的关系；了解素短语最左素短语；理解句柄的条件等。	能够识别短语，简单（直接）短语，句柄，素短语，最左素短语；理解和掌握句柄的条件；理解句柄在移进-归约语法分析中的作用等。				
CP04_08_02	最左归约和句柄剪枝	最左归约，最右直接推导，最右推导（规范推导），规范句型，句柄剪枝等。	理解最左归约，最右直接推导，最右推导（规范推导），规范句型，句柄剪枝等。	理解最左归约、自下而上生成语法树在自下而上语法分析中的运用。	能够运用最左归约、自下而上生成语法树来描述符号串的自下而上语法分析过程。				
CP04_08_03	移进-归约语法分析技术	移进-归约语法分析技术，移进-归约语法分析中的冲突等	了解移进-归约语法分析技术的主要思想。	理解移进-归约语法分析技术，移进-归约语法分析中的冲突；能够描述符号串的移进-归约语法分析过程。	能够运用移进-归约语法分析技术，进行语法分析。				
CP04_09	优先分析方法	优先分析方法的主要思想，存在的问题，解决方案，两种优先分析技术的特点和局限等。	了解优先分析方法的主要思想。	理解优先分析方法、存在的问题和解决方案。	能够运用优先分析方法，描述符号串的语法分析过程；理解两种优先分析技术的特点和局限等。	0.5	0.5	0	

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_09_01	简单优先分析方法	简单优先分析方法，简单优先文法，简单优先分析表，简单优先分析技术的特点和局限等。	了解简单优先分析方法的主要思想。	理解简单优先分析方法的主要思想和条件；理解简单优先文法，简单优先分析表的构造。	理解简单优先分析技术的特点和局限；能够运用简单优先分析方法，描述符号串的语法分析过程。				
CP04_09_02	算符优先分析方法	算符优先分析方法，算符优先文法，算符优先分析表，算符优先分析技术的特点和局限等。	了解算符优先分析方法的主要思想。	理解算符优先分析方法的主要思想和条件；理解算符优先文法，算符优先分析表的构造。	理解算符优先分析技术的特点和局限；能够运用算符优先分析方法，描述符号串的语法分析过程。				
CP04_10	LR 分析方法	LR分析方法的主要思想，LR分析器；识别规范句型活前缀的DFA四种（LR(0)，SLR(1)，LR(1)，LALR(1)）LR分析表，LR分析算法和过程等；四种LR分析方法的特点和局限等。	了解LR分析方法的主要思想，LR分析器的条件；理解LR文法，LR分析过程。	理解和掌握LR分析方法，理解LR分析器，识别规范句型活前缀的DFA；掌握LR分析算法和过程，能够构造LR分析表（LR(0)，SLR(1)，LR(1)，LALR(1)）等。理解四种LR分析方法的特点和局限等。	能够采用LR分析方法进行语法分析；能够构造文法的LR分析表，设计和实现LR分析器。	14	8	6	

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_10_01	LR(k)文法	LR(k) 文法，了解LR(k)文法，LR(k)语言，非LR(k)语言的定义、LR(k)文法，二判定、应用等。义性文法等。		理解和掌握LR(k)文法能够熟练掌握LR(1)文法的判定；LR(k)语言的定义、判定；理解LR(k)文法，非LR(k)文法，二义性文法等。	能够熟练掌握LR(1)文法的判定；并能用LR(1)文法描述LR(1)语言。				
CP04_10_02	LR分析器	LR分析器，了解LR分析器的结构和工作原理。析表，符号栈，LR分析程序等。		理解和掌握LR分析器的结构和工作原理；(LR(0) ， SLR(1) ， LR(1) ， LALR(1)) ，对程序设计语言的基本状态栈，LR分析程序本语法成分进行语法分析，手工等；能够正确描述符构造语法分析器，并对其优势和号串的LR分析过程。局限性进行分析评价。	能够选择合适的LR语法分析技术				
CP04_10_03	LR分析算法	LR分析过程，了解LR分析方法，LR分析算法等。		理解和掌握LR分析方法，LR分析算法等。	能够设计和实现LR分析算法。				
CP04_10_04	LR分析表	LR分析表，理解LR分析表，了解LR分析表的构造。ACTION表，GOTO表，LR分析表的构造冲突，移进-归约冲突，归约-归约冲突等。		掌握LR分析表的构造理解冲突，能用LR分析表进行语法分析。	能够设计和实现构造LR分析表的算法，编程实现LR分析表的自动构造。				

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_10_05	规范句型活前缀	规范句型活前缀/可行前缀, 可归前缀等。	了解规范句型活前缀/可行前缀, 可归前缀。	理解规范句型活前缀/可行前缀, 可归前缀。	理解规范句型活前缀/可行前缀, 可归前缀在LR分析中的作用。				
CP04_10_06	LR(0)项目	LR(0)项目, 移进项目, 待约项目, 归约项目, 接受项目等。	了解LR(0)项目, 移进项目, 待约项目, 归约项目, 接受项目等。	理解LR(0)项目, 移进项目, 待约项目, 归约项目, 接受项目等。	理解LR(0)项目, 移进项目, 待约项目, 归约项目, 接受项目等在表示句柄的识别过程(状态)中的作用。				
CP04_10_07	LR(0)项目集规范族	LR(0)项目集, 规范句型活前缀的识别状态, LR(0)项目集规范族等。	了解LR(0)项目集, 规范句型活前缀的识别状态, LR(0)项目集规范族等。	理解和掌握LR(0)项目集, 规范句型活前缀的识别状态, LR(0)项目集规范族的表示和含义; 掌握GOTO, CLOSURE的求法。	能够用LR(0)项目集表示规范句型活前缀的识别状态; 能够构造LR(0)项目集规范族。				
CP04_10_08	识别规范句型活前缀的DFA	识别规范句型活前缀的DFA LR自动机, 构造思想, 理解该LR自动机的识别过程。	了解识别规范句型活前缀的DFA LR自动机的构造思想, 理解该LR自动机的识别过程。	理解和掌握规范句型活前缀的DFA的作用、构造方法、识别过程。	熟练掌握构造识别规范句型活前缀的DFA的方法; 能够手工构造/编程实现该LR自动机。				

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_10_09	LR(0) 分析方法	LR(0) 文法，LR(0)分析表，冲突和局限等	了解LR(0)文法的定义、判定和应用；理解采用LR(0)分析表进行LR(0)分析的过程。	理解和掌握LR(0)文法的定义和判定；掌握LR(0)分析表的构造；理解冲突（移进-归约冲突，归约-归约冲突和LR(0)分析法的局限等。	能够设计和实现构造LR(0)分析表的算法，编程实现LR(0)分析表的自动构造。				
CP04_10_10	SLR(1)分析方法	SLR(1) 文法，SLR(1)分析表冲突和局限等	理解LR(0)分析法的局限；了解SLR(1)文法的定义、判定和应用；理解采用SLR(1)分析表进行SLR(1)分析的过程。	理解和掌握SLR(1)文法的定义和判定；掌握SLR(1)分析表的构造；理解冲突和SLR(1)分析法的局限等。	能够设计和实现构造SLR(1)分析表的算法，编程实现SLR(1)分析表的自动构造。				
CP04_10_11	LR(1)项目	LR(1)项目，移进项目，待约项目，归约项目，接受项目同心项目等。	了解LR(1)项目，核心项目，同心项目；移进项目，待约项目，归约项目，接受项目等。	理解LR(1)项目，向前看符lookahead的作用等。	理解LR(1)项目，移进项目，待约项目，归约项目，接受项目等在表示句柄的识别过程（状态）中的作用。				
CP04_10_12	LR(1)项目集规范族	LR(1)项目集，规范句型活前缀的识别状态LR(1)项目集规范族等。	了解LR(1)项目集，规范句型活前缀的识别状态LR(1)项目集规范族等。	理解和掌握LR(1)项目集，规范句型活前缀的识别状态，LR(1)项目集规范族的表示和含义；掌握GOTO，CLOSURE的求法。	能够用LR(1)项目集表示规范句型活前缀的识别状态；能够构造LR(1)项目集规范族。				

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_10_13	LR(1) 分析方法	LR(1) 文法，LR(1)分析表，冲突和局限等	理解SLR(1)分析法的局限；了解LR(1)文法的定义、判定和应用；理解采用LR(1)分析表进行LR(1)分析的过程。	理解和掌握LR(1)文法的定义和判定；掌握LR(1)分析表的构造；理解冲突和LR(1)分析法的局限等。	能够设计和实现构造LR(1)分析表的算法，编程实现LR(1)分析表的自动构造。				
CP04_10_14	LALR(1) 分析方法	LALR(1)文法，LALR(1)分析表，合并同心项目，冲突和局限等	理解LR(1)分析法的局限；了解合并同心项目集，了解LALR(1)文法的定义、判定和应用；理解采用LALR(1)分析表进行LALR(1)分析的过程。	理解和掌握LALR(1)文法的定义和判定；掌握LALR(1)分析表的构造；理解冲突和LALR(1)分析法的局限等。	能够设计和实现构造LALR(1)分析表的算法。				
CP04_10_15	二义性文法的LR分析	二义性文法，二义性文法的冲突消解，无二义性规则，二义性规则，用优先级和结合性等解决冲突，二义性文法的LR分析等	理解二义性文法；了解无二义性规则，二义性文法的LR分析。	理解二义性文法，二义性文法的冲突消解；理解无二义性规则，能用优先级和结合性等解决冲突，进行二义性文法的LR分析。	能够通过二义性规则的使用，消解冲突，设计和实现二义性文法的LR分析。				

标识符	名称	说明	能力要求			学时要求			
			了解	掌握	综合应用	总	理论	实践	其他
CP04_11	语法分析器自动生成	语法分析器的自动生成，语法分析器的自动生成器，Yacc等。	了解Yacc、JavaCC等语法分析器生成工具。	了解以LL文法和LR文法为基础的语法分析器生成工具的工作原理。	了解语法规则描述文件的格式，掌握为特定语法规则编写文法规则描述文件的方法；并能够选择和使用一种工具自动生成语法分析器。	2.5	0.5	2	
CP04_12	语法错误和错误恢复	语法错误类型，语法错误的处理，错误恢复等。	了解语法错误类型。	了解语法错误的处理，错误恢复方法等。	能够将错误恢复方法，运用到语法分析器的设计和构造中，优化和完善语法分析器。	0.5	0.5	0	
CP05	语义分析及中间代码生成	语义分析，中间代码生成等	了解描述语言语义的方法；了解典型的中间代码形式；了解语义分析、中析、中间代码生成的基本原理和步骤。	深入理解属性文法，掌握语法制导翻译技术；深入理解语义分析、中析、中间代码生成的基本原理和步骤；掌握语义分析和中间代码生成器的工作原理。	掌握自顶向下、模块化等典型方案；了解语义分析的几种实施方法；了解和学习如何恢复和处理语义分析中遇到的错误；并能够构造一个语义分析及中间代码生成器。	18	12	6	
CP05_01	语义分析	语义分析，静态语义检查，静态语义错误，语义的形式化表示，语法制导翻译技术等	了解语义分析的任务、作用和语义分析器的工作原理。了解语法制导翻译技术的主要思想。	掌握语义分析的基本思想；理解语法制导翻译技术；理解语义分析器的工作原理。	能够选择合适的语义分析技术/方案，完成语法制导翻译工作。	2	2	0	

4

课程标准附录案例

编译原理课程规范

Compiler Principles

框 架

CONTENTS

1. 课程目标与总体要求
2. 课时安排
3. 考核要求



课程目标与总体要求

Specification for Core Courses of Computer Science
Compiler Principles

01

课程目标与总体要求

Compiler Principles

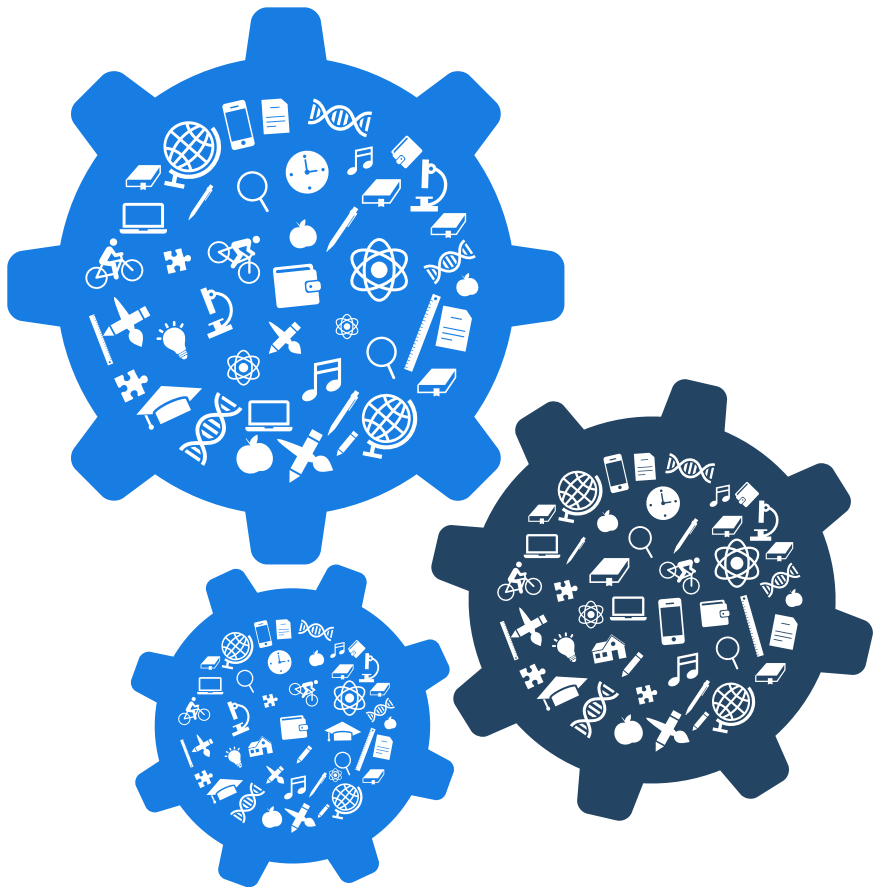
编译原理课程规范

Compiler Principles

框 架

CONTENTS

1. 课程简介
2. 课程定位
3. 课程目标
4. 授课对象
5. 课程学时
6. 前导课程
7. 后续课程
8. 教学要求



课程简介

本课程以编译系统的总体结构为主线，选择语言描述、词法分析、语法分析、语义分析及中间代码生成作为主要内容，介绍编译系统设计与实现及其相关的方法和原理。

课程定位

本课程是计算机专业的专业基础必修课，可以作为其它计算机（类）专业的选修课，属于软件技术系列，是一门理论与实践紧密结合的课程。课程旨在继程序设计、数据结构与算法等课程后，引导学生在系统级上再认识程序和算法，培养其计算思维、算法设计与分析、程序设计与实现、计算机系统认知和应用四大专业基本能力。

课程目标

使学生掌握“编译原理”中的基本概念、基本理论、基本方法，在系统级上再认识程序和算法，提升计算机问题求解的水平，增强系统能力，体验实现自动计算的乐趣。帮助学生了解和熟悉“问题、形式化描述、计算机化”这一典型的问题求解过程，理解和掌握编译过程中所涉及的基本理论和方法，具备分析和实现编译器的基本能力；并培养学生的系统能力和面向系统构建的交流和团队协作开发的能力。

按学历分类

本科生（规范参考主体）

研究生

专科生

按培养目标分类

计算机科学研究（科学型）人才

计算机（类）专业的工程型人才（规范参考主体）

计算机（类）专业的应用型人才

授课对象

按照教育部高等学校计算机科学与技术教学指导委员会发布的专业发展战略研究报告，计算机科学与技术专业（简称计算机专业）培养科学性、工程型、应用型人才。各个学校可以根据自己所在学科、教师、学生的特长以及社会的需求，明确培养目标，并按照确定的培养目标设置课程体系，开展教学活动。

本课程规范的教学设计主要针对培养计算机（类）专业的工程应用型人才的需要制定，并按照理论授课54学时，实践环节36学时进行规划。面向其他类型学生培养、不同学时的教学设计可以酌情参照给出。

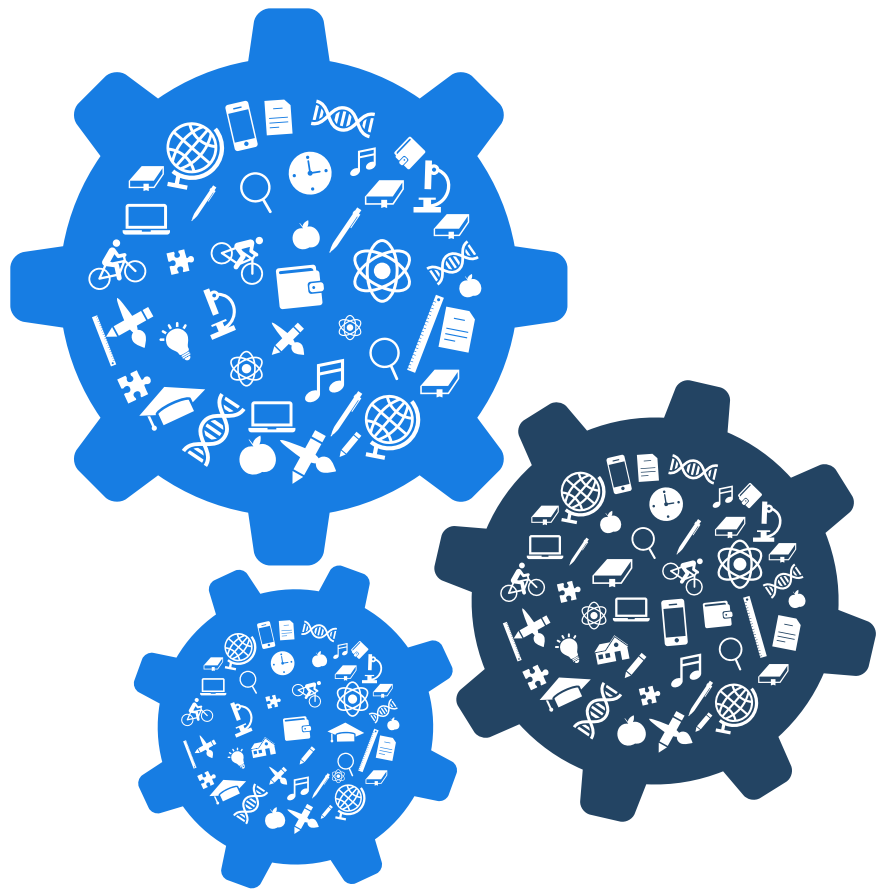
例如，对于未来从事计算机科学研究（科学型）的学生，授课中应该进一步突出对基本原理的研究；对工程类的学生，可以对后面章节多安排学时。特别是如果有更多的学时，建议用于后面章节的学习。当然，要想有更好的理解和掌握，安排一个课程设计是很有意义的。

而计算机（类）专业的工程应用型人才需要具有较扎实的基础理论、专业知识和基本技能，具有良好的可持续发展能力。所以特别强调编译原理中抽象和设计形态的内容，淡化推导等理论形态的内容，着力培养学生的系统能力，以及理论结合实际的能力，而且强调对学科基本特征的体现。

01

课程目标与总体要求

Compiler Principles



课程学时

总学时54，理论授课54学时，课后实践/上机36学时。

前导课程

高级程序设计语言、离散数学、数据结构、计算机组成原理等。

后续课程

编译器/解释器构造、程序设计语言（实现）、形式语言与自动机等。

教学要求

通过讲授、讨论、实验（实践）和作业，使学生对形式语言有初步了解，并能对编译程序的整个结构有较清楚的认识，熟悉和掌握几种主要的编译方法。掌握编译的词法分析、语法分析、语义分析和中间代码生成、代码优化和目标代码生成等阶段的基本理论、方法和技术；并能综合运用系统设计方法、程序设计方法，完成一定规模编译系统/系统核心构件的设计与实现。



课时安排

Specification for Core Courses of Computer Science
Compiler Principles

02 课时安排

Compiler Principles



授课课时安排

对需要课堂讲授的教学内容进行课时安排。



其他课时安排

对讲课之外的教学内容进行课时安排。

授课课时安排

Compiler Principles

知识单元	授课课时	实践学时	涵盖知识点情况	备注
CP01. 语言及其处理	4		CP01_01 , CP01_03 , CP01_04 CP01_05 , CP01_06 , CP01_07	
CP02. 语言及其描述	10	4	CP02_01 , CP02_02 , CP02_03 CP02_04 , CP02_05 , CP02_06	布置语言定义/文法 分类器实习任务
CP03. 词法分析	4	10	CP03_01 , CP03_02 , CP03_03 CP03_04 , CP03_05	布置词法分析实习 任务
CP04. 语法分析	16	16	CP04_01 , CP04_02 , CP04_03 CP04_04 , CP04_05 , CP04_06 CP04_07 , CP04_08 , CP04_10	布置语法分析实习 任务
CP05. 语义分析及中间代码 生成	12	6	CP05_01 , CP05_02 , CP05_03 CP05_04 , CP05_05	
CP06. 运行环境	4		CP06_01 , CP06_02	
CP07. 代码优化	3		CP07_01 , CP07_02 , CP07_03	
CP08. 目标代码生成	1		CP08_01 , CP08_02	
合计	54	36		

为避免课时分解过细，这里仅以知识单元进行课时分配。具体知识点的授课课时安排可依据具体的教学计划和方案酌情制定。表中也列出各知识单元一般必须涵盖的知识点，没有涵盖的知识点应由学生自行进行学习，或者根据课程计划采用选讲、线上学习和讨论等方式学习。

其他课时安排

Compiler Principles

课时类型	内容	对应能力点	要求	课时	备注
实习/ 实践	语言的选择和描述	CPC01_01	选择或自行定义编译器的源语言，并给出该语言的描述。确定语言的语法成分；用EBNF/文法等表示该语言词法规则、语法规则；非形式化描述语言的语义规则。	2	
	构造Chomsky文法分类程序	CPC01_02	熟悉和掌握文法的形式定义及各成分的含义；正确识别文法的类型。构造Chomsky文法分类程序。	2	选做
	词法分析实习	CPC02_01 CPC02_02 CPC02_03	深入理解词法分析的基本原理和步骤；掌握词法扫描器的工作原理，能够手工构造词法分析器，或者选取合适的词法分析器生成工具，自动生成词法分析器。选择充分的测试用例进行测试和调试，对程序设计语言的源程序进行词法分析。	10	可根据教学计划，手工构造和自动生成二选一。
	语法分析实习	CPC03_01 CPC03_02 CPC03_03 CPC03_04 CPC03_05	深入理解语法分析的基本原理和步骤；掌握语法分析器的工作原理，能够采用自顶向下或者自底向上方法手工构造语法分析器，或者选取合适的语法分析器生成工具，自动生成语法分析器。选择充分的测试用例进行测试和调试，对程序设计语言的源程序进行语法分析。	16	可根据教学计划，手工构造和自动生成二选一。

其他课时安排，主要涉及非讲授环节之外的教学内容的安排。根据教学计划中包括习题、实验、实习/实践、上机的教学课时涉及的教学内容进行安排。表中给出了每项教学内容针对的能力点和教学要求等。可根据对学生能力培养的具体要求，适当增加或调整学生自行上机、实习实践的课时要求。

能力单元	能力点名称	能力点要求	能力点类型	涵盖知识点	备注
CPC03. 语法分析技术及 语法分析器的构造	CPC03_01 语法规则的形式化描述	正确理解语言、语言的语法；掌握语言的语法规则的形式化表示（上下文无关文法）；能够理解程序设计语言的文法（如BNF描述），并能对基本的语法结构进行形式化表示。	思维能力点 创新能力点	CP02_06 CP04_02	对比CPC02_01
	CPC03_02 语法分析方法的选择	掌握自顶向下和自底向上语法分析方法的主要思想、特点、异同和适用范围。能评价各种分析方法的优缺点，并根据文法和语言的特点，选择合适的语法分析技术和语法分析的求解模型（下推自动机），采用相应的语法分析算法和语法分析器结构完成语法分析工作。	思维能力点 设计能力点 创新能力点	CP04_04 , CP04_08 , CP04_06 , CP04_07 , CP04_09 , CP04_10	旧版本草案 给出了能力点 对比CPC03_04
	CPC03_03 自顶向下语法分析器 的手工构造	掌握自顶向下语法分析的基本思想；熟悉语法分析器的工作原理、逻辑结构和分析算法；理解和掌握LL(1)分析方法和递归子程序法；熟悉预测分析器的总体结构和预测分析表的手工构造。能够实现简单算术表达式的递归下降分析器。能够选择合适的自顶向下语法分析技术对程序设计语言的基本语法成分进行语法分析，手工构造语法分析器并对其优势和局限性进行评价分析。	思维能力点 实践能力点 设计能力点 创新能力点	CP04_05 CP04_04 CP04_06 CP04_07 CP04_12	
	CPC03_04 自底向上语法分析器 的手工构造	掌握自底向上语法分析的基本思想；熟悉语法分析器的工作原理、逻辑结构和分析算法；理解优先分析方法的主要思想及其局限；理解和掌握LR分析方法及其优势；熟悉LR分析器的总体结构和LR分析表的手工构造。能够理解算符优先分析方法。能够选择合适的LR语法分析技术（LR(0)，SLR(1)，LR(1)，LALR(1)），对程序设计语言的基本语法成分进行语法分析，手工构造语法分析器，并对其优势和局限性进行分析评价。	思维能力点 实践能力点 设计能力点 创新能力点	CP04_05 CP04_08 CP04_10 CP04_12	对比CPC03_03
	CPC03_05 语法分析器的自动构造	了解Yacc、JavaCC等语法分析器生成工具；了解以LL文法和LR文法为基础的语法分析器生成工具的工作原理；了解语法规则描述文件的格式，掌握为特定语法规则编写文法规则描述文件的方法；并能够选择和使用一种工具自动生成语法分析器。	思维能力点 实践能力点 设计能力点	CP04_11 CP04_02	对比CPC03_03 CPC03_04，选修



考核要求

Specification for Core Courses of Computer Science
Compiler Principles

03

考核要求

Compiler Principles

课内考核环节的安排

期末考试方式



课内综合实践环节的进度安排

总评成绩构成

课内考核环节

Compiler Principles

教师可以根据教学计划进行适当调整。学时多、对实践能力要求高的，可以适当增加课内考核的次数和验收标准。反之，可以酌情减少课内考核环节和要求。

但由于编译原理课程的特殊性，一般来说，如果学生不动手设计实现编译器的某些关键模块，在“做中学”的话，是很难真正掌握好课程的重要知识点，达到能力提升的目的。因此，词法分析和语法分析单元，建议进行课内考核，验收学生词法分析模块和语法分析模块的设计（实现）结果。并作为平时成绩的一个重要组成。

考核环节名称	考核环节内容	考核知识单元	预计考核时间	考核形式	备注
词法分析验收	词法分析报告/程序验收	CP03	第8-10周	实验报告/程序 面对面验收	实验报告必选 程序验收可选。
语法分析验收	语法分析报告/程序验收	CP04	第14-15周	实验报告/程序 面对面验收	实验报告必选 程序验收可选。

课内综合实践环节

Compiler Principles

综合实践环节名称	综合实践环节内容	考核能力单元	预计安排时间	阶段性成果	备注
编译器构造/解释器构造	教师讲解课程设计实验方案和进度要求，学生根据要求开始收集资料、组队、分工，并制定初步实验计划。		1-2周	小组分工	
	学生确定要实现的源语言和开发环境，并学习和了解。	CPC01	3-4周	源语言开发环境	
	学生完成【语言及其描述】的实验并提交实验报告；教师组织学生进行课堂讨论1（语言及其处理），并对学生的问题进行答疑和指导。	CPC01	5周	第一部分实验报告	
	学生对编译器总体结构进行设计，定义接口。	CPC06	6周	编译器结构设计	
	学生完成【词法分析】的实验并提及词法分析器和实验报告；教师进行批改、评分和反馈指导。	CPC02	7-8周	词法分析程序&实验报告	
	学生完成【语法分析】的实验并提及语法分析器和实验报告；教师进行批改、评分和反馈指导。	CPC03	9-13周	语法分析程序&实验报告	
	教师组织学生进行课堂讨论2（编译器自动生成工具）和课堂讨论3（自顶向下 VS. 自底向上）并收集和整理学生的共性问题，进行集中答疑和指导。	CPC03	13-14周		
	学生完成【语义分析和中间代码生成】、【目标代码生成】（选做）的实验并提交语义分析和中间代码生成器、目标代码生成器和实验报告；教师进行批改、评分和反馈指导。	CPC04 CPC05	14-18周	语义分析和中间代码生成程序&目标代码生成程序/实验报告	部分选做
学生完成完整的编译器构造，联调、测试，生成最终提交版本；教师组织学生进行小组验收和答辩，评分、生成评价表并反馈。	CPC06	18周	完整的编译器		

为帮助学生全面系统地学习和掌握编译过程中的关键技术，应该采用贯穿整个编译过程的综合性课程设计方案。有些学校为《编译原理》课程配备了专门的实验或者课程设计课程，一般在《编译原理》课程之后开设。这样，可以通过相关完整的编译器/解释器的构造实践，加深对理论知识的理解和掌握，并动手实践并提升构造复杂的系统软件的能力。如果教学计划中没有配套的实验/课程设计课程，这部分的教学任务则需要整合到《编译原理》课程的课内实习或者课内综合实践环节来完成。在同一个学期内，配合课堂理论授课内容的进度来完成。

期末考试

Compiler Principles

期末考试形式为笔试，一般以闭卷方式进行。



课程成绩评定方法

Compiler Principles

课程成绩构成：期末笔试成绩、平时课后作业成绩、随堂测试成绩、课程实习成绩/课程设计验收成绩。

《编译原理》课程实习/课程设计是对学生综合运用编译的基本原理、基本方法和问题的形式化描述、系统设计方法、程序设计方法等完成较大规模系统设计与实现能力的检验。通过对实验系统的设计实现质量的考核和评价，帮助学生和教师了解对《编译原理》课程内容的掌握情况，并为学生的综合能力评估提供参考。

标准方案中，包含了若干课程设计实验和课堂讨论等多种形式，在课程实验中，也在各章节按照循序渐进的原则设置了多个实验。教师和学生应根据实际情况，选择和课程进度相应的，难度适当的实验题目来完成。在最后的课程设计验收时，综合考虑各部分的完成情况，进行客观的评价，特别应关注课程目标的达成评价。

5

后续工作

征询和征集

Email: zmdu@whu.edu.cn

01

课程规范框架、内容和附录部分形式和内容等，征求与会专家和老师们意见和建议；

02

研讨教学实施过程中的关键问题，并体现在标准中；

03

广泛征集编译课程案例、教学设计和课程教学研究报告等，有代表性的作为标准资料性附录；

04

征集附录B（本科编译课程规范案例）

征集附录C（专科编译课程规范案例）



意见和建议

Thank You

Specification for Core Courses of Computer Science

课程标准工作汇报及问题探讨 ←

Email: zmdu@whu.edu.cn