课程详述

COURSE SPECIFICATION

以下课程信息可能根据实际授课需要或在课程检讨之后产生变动。如对课程有任何疑问，请联系授课教师。

The course information as follows may be subject to change, either during the session because of unforeseen circumstances, or following review of the course at the end of the session. Queries about the course should be directed to the course instructor.

1.	**课程名称 Course Title**	**编译原理 Compilers**
2.	**授课院系 Originating Department**	计算机科学与工程系 Department of Computer Science and Engineering
3.	**课程编号 Course Code**	CS323
4.	**课程学分 Credit Value**	3
5.	**课程类别 Course Type**	专业选修课 Major Elective Courses
6.	**授课学期 Semester**	秋季 Fall
7.	**授课语言 Teaching Language**	中英双语 English & Chinese
8.	**授课教师、所属学系、联系方式（如属团队授课，请列明其他授课教师） Instructor(s), Affiliation& Contact （For team teaching, please list all instructors）**	刘烨庞，助理教授，计算机科学与工程系，liuyp1@sustech.edu.cn Yepang Liu, Assistant Professor, Department of Computer Science and Engineering, liuyp1@sustech.edu.cn
9.	**实验员/助教、所属学系、联系方式 Tutor/TA(s), Contact**	
10.	**选课人数限额(可不填) Maximum Enrolment （Optional）**	50

11.	授课方式 Delivery Method	讲授 Lectures	习题/辅导/讨论 Tutorials	实验/实习 Lab/Practical	其它(请具体注明) Other（Please specify）	总学时 Total
	学时数 Credit Hours	32		32		64

12.	先修课程、其它学习要求 Pre-requisites or Other Academic Requirements	(1) The students should be familiar with at least one modern programming language. (2) The students should be familiar with Unix/Linux environment. They should be able to develop, test and debug large software systems. They should also have prior experiences with assembly language. (3) Knowledge of data structures and discrete mathematics is also useful. Prerequisite courses: CS102A 计算机程序设计基础 A Introduction to Computer Programming A or CS205 C/C++程序设计 C/C++ Program Design CS202 计算机组成原理 Computer Organization
13.	后续课程、其它学习规划 Courses for which this course is a pre-requisite	
14.	其它要求修读本课程的学系 Cross-listing Dept.	

教学大纲及教学日历 SYLLABUS

15. 教学目标 Course Objectives

本课程介绍计算机编程语言的基本原理与编译器设计的理论基础。课程将会介绍编译器的各个组成部分，内容主要包括词法分析、语法分析、语义分析、中间代码生成、运行时刻环境、代码生成以及机器无关优化等。本课程还将通过实验使学生了解如何设计并且实现一个简单语言的编译器。

In this course, we will introduce the basic principles of programming languages and theories of compiler construction. The course aims to help students understand the architecture of a modern compiler and will cover lexical analysis, syntax analysis, semantic analysis, intermediate-code generation, run-time environments, code generation and machine-independent optimizations. We will design experiments to provide students hands-on experience of constructing a compiler for a simple language.

16. 预达学习成果 Learning Outcomes

在课程结束时，学生应该获得以下技能：

(1) 了解计算机编程语言的基本原理

(2) 掌握现代编译器的架构及原理

(3) 掌握词法分析、语法分析、代码生成等编译关键技术

(4) 了解代码优化技术

(5) 设计一个简单语言并借助工具实现其编译器

On the completion of this course, students should be able to:

(1) Understand the basic principles of computer programming languages

(2) Understand the architecture of a modern compiler its underlying theories

(3) Understand the key techniques in lexical analysis, syntax analysis, and code generation steps

(4) Understand code optimization techniques

(5) Design a simple language and construct a compiler for the language using tools

17. 课程内容及教学日历 （如授课语言以英文为主，则课程内容介绍可以用英文；如团队教学或模块教学，教学日历须注明主讲人）
Course Contents (in Parts/Chapters/Sections/Weeks. Please notify name of instructor for course section(s), if this is a team teaching or module course.)

Syllabus for Lectures (理论课教学内容安排)：

Chapter 1 (Week 1): Introduction to Compilers (编译器简介)

- What is a compiler? (什么是编译器？)
- The structure of a modern compiler (编译器的结构)
- The evolution of programming languages and compilers (程序设计语言与编译器的发展历程)
- Applications of compiler technology (编译技术的应用)
- Programming language basics (程序设计语言基本原理与概念)

Chapter 2 (Weeks 2-3): Lexical Analysis (词法分析)

- The role of a lexical analyzer (词法分析器的作用)
- Specification of tokens (词法单元的规约)
- Recognition of tokens (词法单元的识别)
- Lexical-analyzer generators (词法分析器生成工具)
- Finite automata (有穷自动机)

Chapter 3 (Weeks 4-5): Syntax Analysis (语法分析)

- Parsers and grammars (语法分析器和文法)
- Context-free grammars (上下文无关文法)
- Top-down parsing (自顶向下语法分析)
- Bottom-up parsing (自底向上语法分析)
- Parser generators (语法分析器生成工具)

Chapter 4 (Weeks 6-7): Syntax-Directed Translation (语法制导的翻译)

- Syntax-directed definitions (语法制导定义)
- Evaluations orders for SDD's (SDD 的求值顺序)
- Applications of syntax-directed translation (语法制导翻译的应用)
- Syntax-directed translation schemes (语法制导的翻译方案)
- Implementing L-attributed SDD's (实现 L-属性的 SDD)

Chapter 5 (Weeks 8-10): Intermediate-Code Generation (中间代码生成)

- Intermediate representations (中间表示)
- Three-address code (三地址码)
- Translations of expressions (表达式的翻译)
- Type checking (类型检查)

- Control flow (控制流)
- Backpatching (回填)

Chapter 6 (Weeks 11-12): Run-Time Environments (运行时刻环境)

- Storage organization (存储分配)
- Stack allocation of space (栈分配)
- Heap management (堆管理)
- Introduction to garbage collection (垃圾回收简介)

Chapter 7 (Weeks 13-14): Code Generation (代码生成)

- Issues in the design of a code generator (代码生成器设计中问题)
- The target language (目标语言)
- Addresses in the target code (目标代码中的地址)
- Basic blocks and flow graphs (基本块和流图)
- Optimization of basic blocks (基本块优化)
- A simple code generator (一个简单的代码生成器)
- Register allocation and assignment (寄存器分配)

Chapter 8 (Weeks 15-16): Machine-Independent Optimizations (机器无关的优化)

- The principle sources of optimizations (优化的来源)
- Data-flow analysis (数据流分析)
- Constant propagation (常量传播)
- Partial-redundancy elimination (部分冗余消除)
- Loops in flow graphs (循环的处理)

Syllabus for Lab Sessions (实验课教学内容安排):

Lab 1: Project Introduction and Environment Setup (课程项目简介及开发环境配置)

- A C-like language (类 C 语言的介绍)
- Project and milestones (项目介绍及时间表)
- Programming environment and tool chain (编程环境与工具链)
- MIPS simulator (MIPS 模拟器)

Lab 2: Lexical Analysis Using Flex – Part 1 (用 Flex 进行词法分析–上)

- Introduction of GNU Flex (GNU Flex 简介)
- Writing specifications of a lexical analyzer (编写词法分析器的规约)

Lab 3: Lexical Analysis Using Flex – Part 2 (用 Flex 进行词法分析–下)

- Regular expressions (正则表达式)
- Generating a lexical analyzer (词法分析器生成)

Lab 4: Syntax Analysis Using Bison – Part 1 (用 Bison 进行语法分析–上)

- Introduction of GNU Bison (GNU Bison 简介)
- Writing grammar specifications for Bison (编写语法规约)

Lab 5: Syntax Analysis Using Bison – Part 2 (用 Bison 进行语法分析–下)

- Generating a parser (语法分析器生成)
- Debugging your parser (调试语法分析器)

Lab 6: Project Presentation and Checking 1 (项目汇报及进度检查 1)

Lab 7: Semantic Analysis (语义分析-上)

- Attribute grammar (属性文法)
- Symbol table (符号表)

Lab 8: Semantic Analysis (语义分析–下)

- Type system and type checking (类型系统与类型检查)
- Finding semantic errors (语义错误检查)

Lab 9: Intermediate-Code Generation – Part 1 (中间代码生成–上)

- IR categorization (中间代码分类)
- IR representation (中间代码表示)
- Translating expressions (翻译基本表达式)
- Translating statements (翻译语句)

Lab 10: Intermediate-Code Generation – Part 2 (中间代码生成–下)

- Translating method calls (翻译函数调用)
- Translating arrays and structs (翻译数组和结构体)

Lab 11: Project Presentation and Checking 2 (项目汇报及进度检查 2)

Lab 12: Code Generation – Part 1 (目标代码生成–上)

- MIPS32 assembly language (MIPS32 汇编语言)
- The SPIM simulator (SPIM 模拟器)

Lab 13: Code Generation – Part 2 (目标代码生成–中)

- Instruction selection (指令选择)
- Register allocation: local allocation (寄存器分配：局部分配算法)
- Register allocation: global allocation based on graph coloring (寄存器分配：基于图着色的全局分配算法)

Lab 14: Code Generation – Part 3 (目标代码生成–下)

- MIPS register usage conventions (MIPS 寄存器使用规范)
- Stack management (栈管理)

Lab 15: Code Optimization (代码优化)

- Constant propagation (常量传播)

- Dead code elimination (死代码消除)

Lab 16: The Final Project Presentation and Checking (项目汇报及最终检查)

18. 教材及其它参考资料 Textbook and Supplementary Readings

主要教材：Compilers: Principles, Techniques, & Tools (Second Edition), Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson Education, 2008.

参考书：

1. Parsing Techniques: A Practical Guide (Second Edition), Dick Grune, Ceriel J.H. Jacobs, Springer, 2010.
2. Modern Compiler Implementation in C, Andrew W. Appel, Cambridge University Press, 2004.
3. Modern Compiler Implementation in Java (Second Edition), Andrew W. Appel, Jens Palsberg, Cambridge University Press, 2004.
4. Modern Compiler Implementation in ML, Andrew W. Appel, Cambridge University Press, 2004.
5. 《编译原理实践与指导教程》，许畅，陈嘉，朱晓瑞，机械工业出版社，2015 年。

课程评估 ASSESSMENT

19.

评估形式 Type of Assessment	评估时间 Time	占考试总成绩百分比 % of final score	违纪处罚 Penalty	备注 Notes
出勤 Attendance		5%		
课堂表现 Class Performance				
小测验 Quiz				
课程项目 Projects		50%		
平时作业 Assignments		10%		

期中考试 **Mid-Term Test**				
期末考试 **Final Exam**		35%		
期末报告 **Final Presentation**				
其它（可根据需要改写以上评估方式） **Others (The above may be modified as necessary)**				

20. 记分方式 **GRADING SYSTEM**

☑ A. 十三级等级制 **Letter Grading**
☐ B. 二级记分制（通过/不通过） **Pass/Fail Grading**

课程审批 REVIEW AND APPROVAL

21. **本课程设置已经过以下责任人/委员会审议通过**
This Course has been approved by the following person or committee of authority