

Spring 2010

# CS 790: Optimizing Compilers for Modern Architectures

Meilin Liu

Wright State University - Main Campus, [meilin.liu@wright.edu](mailto:meilin.liu@wright.edu)

Follow this and additional works at: [https://corescholar.libraries.wright.edu/cecs\\_syllabi](https://corescholar.libraries.wright.edu/cecs_syllabi)



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

---

## Repository Citation

Liu, M. (2010). CS 790: Optimizing Compilers for Modern Architectures. .  
[https://corescholar.libraries.wright.edu/cecs\\_syllabi/515](https://corescholar.libraries.wright.edu/cecs_syllabi/515)

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact [corescholar@www.libraries.wright.edu](mailto:corescholar@www.libraries.wright.edu), [library-corescholar@wright.edu](mailto:library-corescholar@wright.edu).

# *Computer Science (CS) 790*

## *Optimizing Compilers for Modern Architectures*

**Spring Quarter 2010**  
**Wright State University**

### **Course Description**

This course studies compiler optimization for modern architectures. Between parsing the input program and generating the target machine code, optimizing compilers perform a wide range of program transformations on a program to improve its performance. In this course we focus on data dependence analysis, program transformations and loop transformations, loop scheduling, and a combination of these optimizing techniques.

### **Lecturer**

Meilin Liu

Office: 353 Russ Engineering Center

Phone: 937-775-5061

Office Hours: 2:00 – 4:00 pm Tuesday/Thursday

Email: [meilin.liu@wright.edu](mailto:meilin.liu@wright.edu)

Web: [www.wright.edu/~meilin.liu](http://www.wright.edu/~meilin.liu)

### **Class**

- Tuesday/Thursday 8:00– 9:15 pm Russ 406

### **Text**

*Optimizing Compilers for Modern Architectures, Rand Allen and Ken Kennedy, Second Printing, Morgan-Kaufman.*

### **Reference**

*High Performance Compilers for Parallel Computing, Michael Wolfe, Addison-Wesley, 1996*

*Engineering a Compiler, Keith Cooper, Linda Torczon, Publisher: Morgan Kaufmann; first edition, 2003.*

Prerequisite: CEG 320 or CEG 420 Computer Architecture or with the permission of instructor

### **Required Work**

Homework	20%
Quizzes	10%
Project	20%
Midterm Exam	20%
Final Exam	30%

### **Grading**

The base scale is: A: 90-100, B: 80-89, C: 70-79, D: 60-69, F: 0-59. This is the highest requirement that will be used. The scales may be lowered or revised if necessary.

### Policies and Notes

- Attendance: Attendance is not required, but recommended. For your own sake, you should not miss any of the classes. If you are not a regular attendee, it will be your responsibility to seek out what material was covered in the lecture and learn it. Most of my exam questions will be taken directly from ideas covered during the lecture, so it greatly helps if you attend!
- I will utilize webCT (wisdom.wright.edu) to post updates to the course, solutions, assignments, announcements, schedule, etc. Get in the habit of checking it regularly.
- If you are going to miss an exam, for any reason, discuss it with me in advance. If it is an emergency situation, please notify me as soon as possible.
- A penalty of 10% deduction each day for late submission of homework will be given and after one week, 0 point will be given.

### Academic Misconduct

In this class, the only way to truly learn the concepts is to do the work yourself. I encourage working with other people on the course concepts. When you begin to do the homework and the projects, do it on your own.

## Schedule

(The schedule may subject to change.)

Week	Contents	Read, Project
1	Architecture Overview, Memory Hierarchy, Cache Organization	Chap. 1
2	Control Dependence, Control Flow Graph, Basic Block	Chap. 7
3	Data Dependence Theory Analysis	Chap. 2
4	Data Dependence Analysis	Chap 3
5	Instruction Scheduling, List Scheduling	Chap 10
6	Dominator Frontier, Static Single Assignment	From Paper. (Project 1 Due)
7	Program Transformations, Loop Transformations	Chap 4
8	Liveness Analysis, Register Allocation	From Paper.
9	Software Pipelining	From Paper.
10	Software Prefetching, Review	Chap. 8 (Project 2 Due)
	<b>Final Exam:</b> Tuesday, June 8th, 2010, 8:00 -10:00 pm.	