OAKLAND UNIVERSITY, School of Engineering and Computer Science
CSE 436/536 Concurrent and Multicore Programming, Winter 2016

Instructor: Yonghong Yan (www.secs.oakland.edu/~yan), yan@oakland.edu
Office: 534 Engineering Center; **Tel:** (248) 370 4087
Office Hours: After class or by appointment
Meeting Time: 3:30 pm – 5:17 pm Monday and Wednesday; Check http://www.oakland.edu/important-dates for other important date/deadline
Class Room: EC 566 from Jan 11

Course Catalog Description:

This course will focus on concepts, theory, design and implementation of concurrent programs for multi-core computers, multi-core programming methodologies. Topics covered include mutual exclusion, memory model and thread-based parallelism, fork-join framework, locks, parallel control flow, concurrent data structures.
4.000 Credit hours
4.000 Lecture hours
Levels: Graduate, Post Bachelor, Graduate Professional Development, Doctoral & Ed Specialist, Undergraduate
Schedule Types: Main Campus Classroom, Lecture
Computer Science & Engineering Division Computer Science & Engineering Department

Requirements:

Minimum requirements include good reasoning and analytical skills and skills of C programming, e.g. the use of macro, pointer, array, struct, union, function pointer, and library for memory allocation and de-allocation (malloc and free). Knowledge or courses of computer architecture (memory hierarchy, cache, virtual address) and data structures will be necessary for preforming well for the class. Knowledge or courses of programming languages and compilers, and operating systems will also help. Familiarity with Linux environment will be important for the assignments.

Course Objectives:

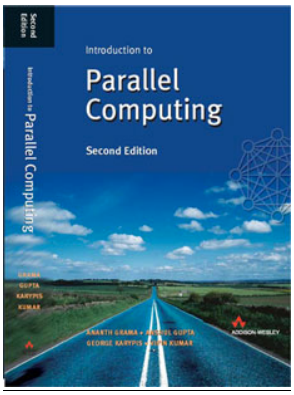
Upon completion of course, successful students shall be able to

1. Describe benefits and applications of concurrent and parallel programming. (a,b,g)
2. Explain key concepts in parallel computer architectures, e.g. shared memory system, distributed system, NUMA and cache coherence. (a,b,c,g,h)
3. Understand principles for concurrent program design, e.g. decomposition of works, task and data parallelism, processor mapping, mutual exclusion, locks. (a,b,c,g, h, I, j, k)
4. Write parallel program using OpenMP, Cilkplus, CUDA, MPI programming models. (a,b,c,d,f,I,j,k)
5. Perform analysis of parallel program problem. (a,b,c,d,f,i,j,k)

Textbook and materials:

Most of the materials can be found from Internet and I will provide a list of links for those resources. There is no **required** textbook. The two recommended textbooks, if you really need ones, are:

Recommended Textbook:



Introduction to Parallel Computing, 2nd Edition
By Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar
Addison-Wesley, 2003
http://www-users.cs.umn.edu/~karypis/parbook/

Grade Assessment:
Course grade will be a combination of 3 homeworks (each 15% for total 45%), one project (50%) and 5% class participation.

Homework and Projects:
In those homeworks, you will apply theory and practice programming skills we discuss during the class. They require both correct programming and good description and discussion in the written report, which should present the approach you take and the results you find in your program. The goal for the project is to investigate specific challenging problem and develop solutions. You will be guided to study related work, identify the problem, develop solutions, perform experiments, analyze your results and report the findings. You will present your project to the class and your report should be in the form of publishable paper. Unless it is announced as teamwork, all assignments and class activities will be completed **individually**.

Behavioral Contract: (For lectures and homework/project)

- This class is offered in classroom with PCs and will include both lecture and lab practices; please read the following carefully, students will be asked to leave the labs if they do not follow this contract:
    - Digital devices such as: cell phones, tablets, ipod, mp3 players… etc need to be placed out of sight and must be set to silent. (I will keep my cell phone avail, for emergency preparedness)
    - No headphones are allowed.
    - Students are expected to use the PCs only for this class's material and applications.
    - Surfing the net for Social sites and being on Social Media are not allowed **during lecture hours**. Anyone who is not following this rule will be asked to leave the laboratory/lecture.

Attendance

- Attendance is required in all of the classes; it is part of a student's grade not an extra credit.
- **If you miss more than 2 classes you will not receive any credit for attendance. (The 2 classes include excused absences.)**
- Class participation is part of your grade make sure to participate in our class practices.

Late Policy: (PLEASE READ CAREFULLY)
Each assignment has a due date and a cutoff date, which is 3 days after due date. You will have a total of 7 "slip days" throughout the semester that you can use at your discretion to turn in programming assignments past the posted due date. Slip days are used in whole day increments. Once your slip days are consumed, late programming assignments will be penalized at 10% per day. Assignment submission will not be accepted after the cutoff date.

Moodle:
A session specific website is located at https://moodle.oakland.edu/moodle. This website will include all the course materials (notes, schedules, assignments, etc) for our course. Assignment will be available for download from this site, and they should be submitted using Moodle only. Please check this site often for updates.

Academic Conduct: *Expected conduct on assignments and exams*
Although students may discuss an assignment, each student should complete his or her assignment individually. Copying of another's assignment is not permitted. It is assumed that ALL work throughout the term is your own. Discussions during an exam or quiz are not permitted. Cheating during an exam or quiz is not permitted. It is assumed that ALL WORK THROUGHOUT THE TERM IS YOUR OWN! **Discussion of lab assignments are permitted but copying of assignments is not! Handing in a lab assignment or exam that was essentially copied from someone else does constitute as cheating**. All of the tests are closed book unless it was told otherwise. Obtaining help from notes, another individual or from hand held computing devices during an exam is regarded as cheating. The Oakland University Academic Conduct Policy can be found at http://www4.oakland.edu/?id=1610&sid=75.Cheating on examinations, plagiarism, falsifying reports/records, and unauthorized collaboration, access, or modifying of computer programs are considered serious breaches of academic conduct. The Oakland University policy on academic conduct will be strictly followed with no exceptions. See catalog under Academic Policies and Procedures.

Tentative class content and schedule:

Week	Date	Week date	Class	Content (Tentative)	Assignment
	01/04	Monday			
1	01/06	Wednesday	1	Introduction	
	01/11	Monday	2	Review of C programming, Compiler, Makefile, Linux and SSH access	Assignment 1 (Advanced C Programming)
2	01/13	Wednesday	3	Parallel algorithm design 1	
	01/18	Monday		MKL Jr. Day, no class	
3	01/20	Wednesday	4	Parallel program design 1 practice	
	01/25	Monday	5	Parallel algorithm design	Assignment 1 due
4	01/27	Wednesday	6	OpenMP	Assignment 2 (OpenMP programming)
	02/01	Monday	7	OpenMP	
5	02/03	Wednesday	8	OpenMP practice and assignment 2	
	02/08	Monday	9	Cilk	
6	02/10	Wednesday	10	Cilk practice and performance analysis	Assignment 2 due
	02/15	Monday	11	Parallel program measurement and analysis, part 1	Assignment 3 (OpenMP and Cilk)
7	02/17	Wednesday	12	Parallel program measurement and analysis, part 2	
	02/22	Monday		Winter Recess	
8	02/24	Wednesday		Winter Recess	
	02/29	Monday	13	PThread	Assignment 3 due
9	03/02	Wednesday	14	PThread and mutual exclusion	Project Ideas and teams
	03/07	Monday	15	Parallel architecture	
10	03/09	Wednesday	16	Parallel architecture and cache coherency	
	03/14	Monday	17	Memory optimization practice	
11	03/16	Wednesday	18	Manycore and GPU/CUDA	
	03/21	Monday	19	Manycore and GPU/CUDA	
12	03/23	Wednesday	20	GPU/CUDA practice and assignment 3	
	03/28	Monday	21	Manycore and Offloading	
13	03/30	Wednesday	22	Distributed memory systems	
	04/04	Monday	23	MPI	
14	04/06	Wednesday	24	MPI practice and assignment 4	
	04/11	Monday	25	PGAS and UPC	
15	04/13	Wednesday	26	Parallel algorithms - dense matrix	
	04/18	Monday	27	Parallel algorithms - sorting	Last class
16	04/20	Wednesday			
	04/25	Monday	28	Project Presentations (3:30 – 6:30PM)	Project Report Due
17	04/28	Thursday		**Grade due and complete**	

Note: Assignment due 11:55PM on the date, cut-off date is 72 hours after the due date/time

Contributions of Course Objectives to Program Outcomes:

a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;

b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;

c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;

d) An ability to function effectively on teams to accomplish a common goal;

e) An understanding of professional, ethical, legal, security, and social issues and responsibilities;

f) An ability to communicate effectively with a range of audiences;

g) An ability to analyze the local and global impact of computing on individuals, organizations and society;

h) Recognition of the need for, and an ability to engage in, continuing professional development;

i) An ability to use current techniques, skills, and tools necessary for computing practice;

j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;

k) An ability to apply design and development principles in the construction of software systems of varying complexity.