# Parallel Programming
# Course Introduction for those Taking it Online

**Mike Bailey**

**mjb@cs.oregonstate.edu**

**Oregon State University**

# What this Course Is

This course is all about parallel programming on the **desktop** for applications that you are attempting to accelerate to improve user interaction and simulation and computational performance

The goals of this course are to leave you **"career-ready"** (i.e., both work-ready and research-ready) for tasks that require desktop parallelism, both on a CPU and a GPU. CS 475/575 topics include:

• Parallel computing: types, limitations
• Moore's Law, Amdahl's Law
• OpenMP
• Synchronization issues in parallel computing
• Cache issues in parallel computing
• SIMD
• GPU computing
• OpenCL
• Xeon Phi

# What this Course Isn't

This course is not about supercomputers or clusters. A lot of the same principles that we will discuss about the desktop do apply to supercomputers and clusters so this will still be useful.

However, if we have time, we will lightly touch on the Message Passing Interface, MPI, which *is* used in supercomputers and clusters.

OSU

# Instructor

This course is being taught by:

**Professor Mike Bailey**
OSU Computer Science
2117 Kelley Engineering Center
mjb@cs.oregonstate.edu
541-737-2542

Welcome! I'm happy to be here. I hope you are too !

Mike has had over 30 years experience in the computer graphics and high performance computing worlds. He has taught over 100 university classes to a total of over 6,000 students. He has also taught over 80 professional short courses around the world.

In his spare time, he … oh, wait, there is none. Well, that's not quite true. He does like to read. In the spare time within all of that, he dabbles in K-12 outreach, particularly where he can apply computer graphics to it. Does that count as "spare time" or "more work"? ☺

# Office Hours and Other Help

Sadly, I am a compulsive email checker.  That is the best way to reach me.  However, I am not necessarily a compulsive email-*returner*.  I prioritize my email returns.  Please do not email me over small things that you really can either look up or figure out for yourself (like "When is the assignment due?"). Those get low priority.  The really serious concept questions get high priority.

I am a compulsive telephone ignorer.  That is not a good way to reach me consistently.  I will return messages, but it might take a while.  Email is better.  This is not meant to be a disrespectful thing.  It's just that I am in and out of my office a lot, and when I am in, there are usually students in there with me.

I will hold Office Hours over the Internet on WebEx.  I have my own "room".  It's URL is:

**https://oregonstate.webex.com/meet/mjboregonstate.edu**

It's only available when I am there to "open it up" (not unlike a real room).  To use this, you will need a microphone and speakers.  I recommend a headset.  They pickup less background noise than a microphone does and it gives you better sound than a speaker does.  I found a nice inexpensive one at Radio Shack.

I would love it if you also have a webcam, because then I can see you and get to know you better.  And, because I often do these Office Hours from home, I sometimes get a little "help" from the following characters …

# … Some Other Characters You Might "Meet"



## Loki

Loki is a 9-year-old 90-pound dog who thinks he is still a puppy. I hope he doesn't jump up on my lap when I'm holding WebEx Office Hours. If he does, I might need to pause to go get medical treatment.



## Bond

Bond is very out-going. When I am on the computer at home, such as when holding WebEx Office Hours, he likes to jump up on my lap and stare into the webcam. He also likes to sit on my pile of tests when I am grading.

## Callie

Callie is very shy. If you came to my house, she would hide under a bed. But, through WebEx, she won't know you are there. You might see her, you might not.

mjb – March 6, 2017

# What You Should Know on the Way In:
## the Course Incoming Expectations

Above all, you should be a good C programmer.  Being comfortable with function calls, arrays, for-loops, structures, arrays of structures, structures of arrays, pointers, and linked lists is a must.  It is strongly suggested that you not use this class as an opportunity to learn C for the first time.

On the math side. You should know algebra.  There will be times when we have an equation that solves for "Y given X" and I will ask, "What if we already know Y, can we then go back and find X?".

# What We Will Be Covering

Note: this schedule is *approximate*!
I will try to keep the schedule on the class web site up-to-date.

| Week | Topics |
|------|--------|
| 1 | Introduction. Syllabus. What this course is ... and isn't.<br>Project notes: timing, graphing. Examples.<br>Parallel programming background information. The three things we care about Parallel Processing for. Von Neumann architecture.<br>Multithreading |
| 2 | Moore's Law. What holds, what doesn't.<br>Multicore. Hyperthreading.<br>Timing. Speedup. Amdahl's Law. Parallel efficiency. Gustafson's Law.<br>OpenMP: fork-join model, pragmas, what it does for you, what it doesn't do for you,<br>OpenMP: parallelizing for-loops<br>OpenMP: variable sharing, dynamic vs. static thread assignment.<br>Chunksize. |
| 3 | Summing. Not doing anything special vs. critical vs. atomic vs. reduction.<br>Trapezoid integration.<br>Mutexes.<br>Barriers.<br>OpenMP: sections, tasks, graph traversal.<br>Caches. Architecture. Hits. Misses. |

# What We Will Be Covering

| Week | Topics |
|:---:|:---|
| 4 | Caches, cont. False sharing. . <br> Designing parallel programs |
| 5 | Tasks. <br> Barriers. . <br> A special kind of parallelism: Single Instruction Multiple Data (SIMD). <br> SSE, AVX, AVX-512 instructions: what they are, how to use them. <br> Types of problems that work this way. <br> Test #1 |
| 6 | Go over the test answers. <br> GPU 101. <br> Architecture. <br> What GPUs are good at. What they are not good at. Why? |

# What We Will Be Covering

| Week | Topics |
|------|--------|
| 7 | OpenCL Reduction.<br>OpenCl Events |
| 8 | OpenCL / OpenGL Interoperability<br>Looking at OpenCL Assembly Language. |
| 9 | Looking at OpenCL Assembly Language.<br>OpenCL / OpenGL Interoperability<br>**Guest Speaker: Patrick Neill or Chris Schultz from NVIDIA: "GPU Architectures** |
| 10 | OpenGL Compute Shaders<br>More Information |

# Class Textbook

There is no textbook for this class. The course material will consist of handouts and notes taken while watching the videos.

If you need further reference material, there are a bunch of links at the end of the class web site. You're not required to go look at any of these. They are just some links that I have found useful. They are there if you need them.

# Canvas vs Web Pages

I would *love* to do this whole class in Canvas-only.  There is a lot to be said for consistency of interface.  But, Canvas has certain things it can't do.

For example, you will use *Teach* (**http://engr.oregonstate.edu/teach**) to turn in your assignments.  Because I can write scripts behind this system to extract parts of your submissions, I can grade them, and give you feedback a lot faster.  Canvas will not let me do that.

So, we will use Canvas for a lot of things, but not all.