CptS 464/564 Syllabus
Distributed Systems Concepts and Programming

Spring, 2019

TuTh 250-405

Pullman **EME B-46**
Tri Cities Room **TFLO 224**
NPS Everett: **Everett-Seminar-254**

version 2, January 23, 2020

Instructors: Prof. David ("Dave") Bakken

Office: EME 55; **Office phone:** N/A (I don't use it) (Bakken) **E-mail:** bakken@wsu.edu ➔ USE Blackboard email, not this directly, for class-related things please.

➔ In a pinch you may call **my cell five-oh-nine 592-0238** if you need to.

Office Hours: TuTh 410-500 in my office (EME 55). Also by email appointment (email me 5 times you could meet) and when needed for specific assignments.

TA: Reza Chowdhury, for programming projects only. Email him via Blackboard if necessary.

Web page: http://www.eecs.wsu.edu/~cs464 (for both 464 AND 564)

Lecture notes, assignments, and other reading will be posted as PDF files. I will not, as a rule, provide hardcopy handouts of the notes. You are expected to take your own notes during class. They are generally online before a lecture, however, linked in from the class web page. This way you can be looking through the slides and take notes only on what is said but not in the slides.

AMS Coordinators

Tri Cities: Aaron Brumbaugh, 509-372-7284**,** AMS.TriCities.WSUTC@wsu.edu
NPS Everett: Alistair Boudreaux, 425-405-1592, alistair.boudreaux@wsu.edu

Lab space: none.

You will use your personal laptop or desktop for programming assignments.

1) Background

In the last three decades, the world wide web has proliferated to a huge degree. Further, the *Internet of Things (IoT)* has all kinds of devices from phones to refrigerators to sensors are on the internet, and a myriad of services and supporting servers are offered, increasingly via cloud services or peer-to-peer infrastructures, to a complex array of client programs. Also, *Cloud Computing* is pushing computations and data that have traditionally been on a workstation and a LAN-based server that it uses out across the internet, into the (underdefined) "cloud". These trends are expected to continue to even larger degrees in the coming years.

This course is about distributed computing, a layer above networking technologies. Computer networking deals with how to get bytes of data from Point A to Point B with some desirable characteristics (delay, reliability, etc). The field of distributed computing, on the other hand, deals with how to use computer networks to support application programs and servers that cooperate in some fashion. That is, **how do we coordinate, synchronized, replicate, spread an application and/or service over many networked nodes, etc**.

In this class we will study distributed computing technologies supporting the issues raised above, and more. This includes how to use middleware, and to a lesser extent, how middleware is built. Middleware provides higher-level building blocks to programmers than network-level interfaces do, and at the same time is designed to shield programmers from many of the complications that arise when different pieces of programs are spread over a network or the Internet. This class will include not just written exercises and exams, but also programming assignments using middleware. It is also a conjoint undergraduate-graduate course, with a joint lecture but grad students being responsible for additional material and will be graded on additional exercise problems, exam questions, and requirements for a given programming assignment. Although distributed algorithms and the fundamentals of distributed computing are definitely studied, this is not a theory class but is rather an applied application-focused class, and one in a very hot area to industry.

2) Course Objectives

As a result of this course, students will:

- Be well-versed in the fundamental issues involved in designing, programming, and using distributed systems
- Be familiar with some of the most significant new software technologies for creating distributed programs
- Have significant experience using middleware and programming and evaluating distributed algorithms.

3) Text

a) Required Texts:

[CDKB5] "Distributed Systems: Concepts and Design", 5ed, Addison Wesley, May 2011, ISBN 0-13-214301-1

b) Optional Texts:

[VR01] Verissímo, Paulo and Rodrigues, Luís. *Distributed Systems for System Architects*, Kluwer Academic Publishers, 2001, ISBN 0-7923-7266-2.

[TvS17] Andy Tanenbaum and Maarten van Steen. *Distributed Systems:3rd edition*, 2017 (self-published); you can get a FREE copy from the author, here.

4) Additional Reading

Additional papers will be either handed out, linked to the web page, or both.

5) Course Prerequisites

Course Prerequisite: CPT S 223 with a C or better, CPT S 233 with a C or better, or E E 234 with a C or better; certified major in Computer Science, Computer Engineering, Electrical Engineering, Software Engineering, or Data Analytics. Also ***fluency and significant experience in C++ or Java***.

If you do not meet these prerequisites, you **MUST** come and talk with me the first week of class. I reserve the right to drop you from the course if it becomes obvious that you do not meet the prerequisites.

6) Course Requirements

You are *required* to attend every lecture. If you miss one, it is *your* responsibility to find out what happened and to collect any material that was handed out in class. This class is being recorded, and is available for both live streaming and also archival viewing later.

You are also expected to participate in class discussions. This aids your learning and that of your classmates, and provides valuable feedback on the lecture. I reserve the right to lower the grade of any student who is markedly deficient in attendance and/or participation.

I expect ***you*** to ***own*** your degree of success in this class. *And*, I expect you to contribute to the success of others. Examples:

1. If a lecture point is unclear, ask a question, either in class, during office hours or by e-mail. You are probably not alone in your confusion. I enjoy engaging in technical conversations with students with the goal of helping them create an accurate understanding of course material. Participating in such conversations is very favorable for your class participation grade. *I hate talking about how you can get a better grade*; lets talk about how you can learn more.
2. If another student is confused, help him or her out.
3. If I am systematically doing something that inhibits your learning, tell me.

You are ***required*** to check your WSU Outlook Live email; we will use the Blackboard system for this class for assignments and it only forwards email to that.

7) Assigned Work and Tentative Grading Policy

There will be two in-semester exams: a midterm and a final; the final exam will only cover the material discussed and assigned after the first exam. There will be 3-4 homework assignments given out. There will be 2 programming projects, all involving C++ or Java and DDS.

The following allocation of grade percentages is *tentative*, and may change during the semester (**the grades in this class are curved**; I like low curves):

Component	
Exams (2):	40%
Homeworks (3-4+) and Surprise Quizzes :	20%
Programming Projects (2):	40%

Exams (all in the lecture room):

Midterm: TBD, but possibly February 20

Final: Monday May 4, 2020, 1010-1200.

In order to receive a grade higher than C- for the course, you must achieve an average grade of at least C on the exams ***and*** an average grade of at least C on the projects. I reserve the right to downgrade the final grade of someone who does much, much better on exams than he/she does on the projects and homework.

564 students will be given additional material to read, as well as additional homework, programming requirements, and exam questions. Accordingly, 464 and 564 will be graded on separate scales.

8) Policies and Expectations

a) Do your own work

Your exams, homeworks, and programming projects are subject to the academic honor code. **DO NOT CHEAT IN ANY WAY: *DO YOUR OWN WORK!*** Side effects of cheating may involve expulsion, revocation of assistantships, etc., not to mention eliminating any chance of your *ever* working with me.

It is quite acceptable to ask others things like "Have you gotten this exception before?," and even have them look briefly at your stack trace and its code Or "have you seen this problem before?". It is quite unacceptable, on the other hand, to have them spend hours helping develop or seriously rearrange your program's logic. And, of course, it is unacceptable for two or more people to collaboratively develop the solution for a project.

If you are tempted to collaborate on projects, **DON'T!** A few years ago someone did serious cheating in my class (and a few others), and that individual ended up being expelled from WSU! Don't start down that slippery slope!

b) Responsibility for learning

1. I expect you to demonstrate critical thinking across the spectrum of course work. Adaptations of the WSU Critical Thinking Rubric will be used in grading some of the projects and homework.
2. I expect you to cooperate with other students and to pull your share in class discussions. Respect that different people in your group may have different ways of learning and different strengths. Seek ways of taking advantage of those differences.
3. I expect you to engage in *active learning*: speak up when you don't understand, question assumptions, relate course material to your experience outside class, seek out additional experience and reading related to the class. *You* must *construct* your understanding of the material.
4. I expect you to promptly review feedback you receive from me, the TA, or other students; to actively clarify the feedback if the material is still unclear; and to incorporate the feedback in your future work.
5. I expect you to spend adequate time on the course. Adequate time includes getting enough rest so that time you spend on course tasks is well-spent time. Adequate time includes proofreading and reviewing your assignments before you hand them in.
6. I expect you to have high expectations of yourself: set goals for yourself and try to do your very best. Consciously think about the balance between what you do to earn a grade and what you do to learn. (If I'm doing something that puts these in opposition to each other *please* let me know.)
7. You are expected to know and observe the WSU Academic Integrity Standards and the EECS Academic Integrity Policy. Copying of other students' work, working together on individual assignments, plagiarism of published sources and other forms of academic dishonesty will result in zero credit on the assignment for all students involved and a lower grade in the class. A second offense (across the University) will result in an automatic F in the course and exposes the violator to University sanctions up to and including expulsion. All offenses will be reported to Student Affairs.

c) Accommodations for disabilities

Reasonable accommodations are available for students who have a documented disability. Please notify me during the first week of class of any accommodations needed for the course. Late notification may cause the requested accommodations to be unavailable. All accommodations must be approved through the Disability Resource Center (DRC) in Administration Annex room 205, 335-1566, drc@mail.wsu.edu.

d) Emergency Info

Please check out the WSU emergency management web site at http://oem.wsu.edu/emergencies.

e) Advice

1. Don't wait until the last minute to do homework or projects. Computers break down, stuff happens, and people get sick. These are not sufficient excuses for an extension.
2. Save early; save often!
3. Come see me if you are confused. Don't wait for office hours -- send email.
4. Come see me anyway. I'd like to know more about you.
5. If you have extenuating circumstances hindering your ability to get assignments done on time, let me know ASAP, not after the fact.
6. If you have trouble with spelling and grammar, get and use a word processor that has spelling and grammar flagging. Your writing for this class should be of professional quality.

9) Tentative Schedule

We will cover 8-10 chapters from [CDKB5], possibly a little from [VR01], plus have additional lectures on DDS and other middleware plus probably a few other topics from research papers. The exact topics will be refined as I get a better feel for the students' backgrounds. But the chapters we will cover for sure are below, with ones we will not cover are designated with ~~strike through~~:

1: Characterizations of Distributed Systems
2: System Models
3: Networking and Internetworking
4: Interprocess Communication
5: Remote Invocation
6: Indirect Communication
~~7: Operating System Support~~
8: (MAYBE) Distributed Objects and Components
~~9: Web Services~~
10: Peer-to-Peer Systems
14: Time and Global States
15: Coordination and Agreement
Possibly, if time allows:
Cloud Computing (supplementary material)
Fog and Edge Computing (supplementary material)

Notice that **cloud computing** is not covered in huge depth here, despite being a very hot topic. Why? Simply put, the above are many of the core technologies that are used to implement cloud computing! We will overview cloud computing for a day or two, however. Also, the chapters in bold above (2, 6, 10) are what I consider the "crown jewels" of [CDKB5] that set it apart from other textbooks.

10) Suicide Resources

Being a student can be stressful. If you or anyone you know are considering suicide, please avail yourself of the many resources at the Pullman campus. Here is a memo from the Provost about all kinds of related resources.