

# CptS 464/564 Syllabus

## Distributed Systems Concepts and Programming

**Fall, 2011**

MW 2:50-4:05PM

Pullman EME B-46

AMS TV in WSU Tri Cities Room 224-West

<http://www.eecs.wsu.edu/~cs464>

version 1, August 22, 2011

**Instructor:** Prof. David (“Dave”) Bakken

**Office:** EME 55; **Office phone:** 335-2399 (Bakken) **E-mail:** [bakken@eecs.wsu.edu](mailto:bakken@eecs.wsu.edu)

**Office Hours:** TBD, starting the second week of class – or email me for an appointment – I’m happy to see you at other times. Students at the AMS sites should feel free to call during office hours or make an appointment for me to call you. I will try to travel to the Tri-Cities at least once during the semester and hold office hours there before or after class. Come to Prof. Bakken with general course questions, to the TA with program assignment details and debugging issues.

**TA:** Dario Alvarez email [dalvarez@eecs.wsu.edu](mailto:dalvarez@eecs.wsu.edu)

**TA Office Hours:** None regularly scheduled. We will have some associated with each assigned programming project.

**Web page:** <http://www.eecs.wsu.edu/~cs464> (up about the second week of classes)

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.

**AMS (formerly WHETS) Coordinators**

**Tri Cities:** Aaron Brumbaugh, 509-372-7284, [AMS.TriCities.WSUTC@wsu.edu](mailto:AMS.TriCities.WSUTC@wsu.edu)

**Mailing List:** TBA. I will have to set something up; be sure that you either read your EECS email address often or you have it forwarded to somewhere that you do read it from.

**Lab space:** none.

Your EECS computer account is used to login to these computers where we have the DDS software installed.

## 1) Background

In the last decade, the world wide web has proliferated to a huge degree. Further, 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. All of 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. 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, specifically the Data Distribution Service (DDS), a standard, cross-platform software framework for creating distributed programs.

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

[TvS07] Andy Tanenbaum and Maarten van Steen. *Distributed Systems: Principles and Paradigms, 2ed*, Prentice Hall, 2007.

## 4) Additional Reading

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

## 5) Course Prerequisites

Cpt. S. 360 (Systems Programming), or equivalent from another university, or permission from the instructor. 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 grades and how you can get a better one.*
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 [Angel](#) system for this class for assignments and it only forwards email to that.

## 7) Assigned Work and Tentative Grading Policy

There will be one mid-term exam and one final examination; 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 three programming projects, all involving C++ (or possibly 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 (3):	40%

Exams (all in the lecture room): **Midterm WEDNESDAY October 12** **Final Wednesday December 14 3:10-5:10pm**. Note also that there is no lecture on Sept. 5 (Labor Day) and during Thanksgiving break.

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.

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. 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!**

### 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](mailto: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. The lab gets busy, computers break down, and people get sick. These are not sufficient excuses for an extension.
2. Save early; save often!

3. Come see me or the TA 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 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.

The chapters we should cover for sure include the following (see Page XIII in the Preface), and we will not cover ones with ~~strike through~~:

### **Foundations:**

- 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~~

### **Middleware**

- 8: Distributed Objects and Components
- ~~9: Web Services~~
- 10: Peer-to-Peer Systems

### **Distributed Algorithms**

- 14: Time and Global States
- 15: Coordination and Agreement (probably not get to)

### **Substantial Case Study**

- 21: Designing Distributed Systems: Google Case Study