Syllabus
SER321 Principles of Distributed Software Systems

1. Instructional Faculty
   Dr Tim Lindquist; mailto:Tim.Lindquist@asu.edu  Course coordinator.

2. Catalog Description
   Design and implementation of distributed software components; process and memory management underlying software applications; sockets, protocols, threads, XML, serialization, reflection, security, and events.

3. Prerequisites
   1. Ser222 Design and Analysis of Data Structures and Algorithms
   2. Pre, or Co-requisite: Ser334 Operating Systems and Networks

4. Course Overview and Suggestions
   There is significant C++ and Java software development required to complete this course. Also, the course is structured so you must use fundamental Linux program development tools for program submission. You may not submit an IDE project as the solution to any assignments in this course. The programming and the use of alternative development and build tools requires a significant effort on the part of students. Despite the formal pre-requisites, students who have not already taken SER221 Programming Languages and their Execution Environment, (in which C/C++ are covered) before taking Ser321 are left with the challenges of learning a new set of development tools and a new Language in the same semester. This is certainly do-able for a Software Engineering major, but budget your time accordingly. For most Software Engineering students, this is the first course that expects students to learn and utilize programming languages without prior formal (in-class) training.

   Principles of Distributed Software Systems is a software development intensive introduction to the applications that run with components on multiple processors (physical or virtual). It is a required course in the BS Software Engineering curriculum and is recommended to be taken at the beginning of the junior year. The course involves programming in both C++ and Java, and you will work in a basic Linux environment. You will be required to purchase and configure a Raspberry Pi computer (about $50) running Debian Linux as your server deployment machine, and you will be expected to run your client software on a Virtual Box Debian Linux virtual machine. The Virtual Box Debian Linux machine will be your program development environment for the course. Please budget 2-3 hours outside of class per class hour to cover time for complete programming assignments and other course work. Budget more time if you are unfamiliar with Linux, Java, or C++. Face-to-Face students generally have two class sessions per week (about 10+ hours per week). Online students generally complete the course in half the calendar time, which generally requires about 20+ hours per week.
If you wish to purchase your Raspberry Pi and/or get started on configuring the Virtual Box Debian machine before the class begins, you will find information about the development and deployment environments for the class, and correspondingly required software at:

http://pooh.poly.asu.edu/Ser321/Resources/tools.html

You will find further course topical information on the Web at:
http://pooh.poly.asu.edu/Ser321/index.html

5. Grading

Final grades will be awarded based on the following weights:
40% Programming Assignments (probably 6 equally weighted assignments),
30% Midterm Exam,
30% Final Exam

If your overall percentage in the course is 90 or better, you’ll get an A; 80 or better, you’ll get a B, and so forth at 10 percentage point breaks.

All assignments will be submitted as archives on the Blackboard site for the course. No late assignments will be accepted. Grade appeals for programming assignments or exams should be submitted in writing (email) within 1 week from the time the assignment or exam grade is returned to the student. Send all grade appeals by email to both the TA/Grader and the instructor. They will not be accepted in any other form. Since assignments may change from semester to semester, the assignment page (web site or Blackboard) may change up until the date the assignment is discussed in class. Before submitting an assignment solution, please extract the archive (jar file) you intend to submit into a clean directory. Then compile and execute the program, from this jar file only. That is, do what the grader will do.

If you are an Online student, you may be required to pay for and use a testing service, to complete exams.

6. Academic Integrity Policy (AIP)

Programming projects in this course are individual and not group projects. Points will be deducted from any assignment that is not turned in with a copyright notice (belonging to the student) and assigning right-to-use to the instructor and ASU for the purpose of student and course evaluation. You may use any part of the open-source instructor examples in completing your assignments. You may not use software obtained from another student, or any other source in completing the assignments. You may use any of the instructor’s examples all of which are marked providing you with the right to use.

Discussions among students on class material and assignments are encouraged, but all students in this course must follow the University Academic Integrity Policy (AIP) and Student Code of Conduct. In doing so, Students must turn in their own independent work in this class. Any student found in violation of AIP or the Student Code of Conduct (linked below) may be given a failing grade for the course. A student may be removed from the program for such violations. Do not represent
someone else's work as your own, and do not let someone else represent your work as their own. Grading may include executing student solutions using software that compares the structure and content of student files. Any cases of suspected violation of AIP are referred to the college office according to established policy. By your registration, you are assumed to understand and agree to this policy.

AIP: https://provost.asu.edu/academic-integrity