Syllabus
SER321 Principles of Distributed Software Systems

1. Instructional Faculty
   Dr Tim Lindquist; mailto:Tim.Lindquist@asu.edu Course coordinator.
   A course leader, aside from the coordinator, may be assigned, and made available when the semester begins.

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 a significant amount of programming in this course in both C++ and Java. The course is structured so you must use fundamental Linux program development tools -- no IDE. This requires a significant commitment on the part of the student. Despite the formal pre-requisites, its recommended to follow the Software Engineering Major-Map and also complete SER221 Programming Languages and their Execution Environment, before taking Ser321. Doing so will ease the burden of the C++ programming required in this course. For most Software Engineering students, this is the first course (of several) that expect 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 machines. 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, and will do your program development on that environment. Please budget 2-3 hours outside of class per class hour to cover time for complete programming assignments and other studying time. 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 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 (6 equally weighted assignments),
- 30% Midterm Exam,
- 30% Final Exam

All assignments will be submitted as archives on the Blackboard. 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 is returned. Send to both the TA and the instructor. Since assignments may change from semester to semester, the assignment page may change up until the date the assignment is discussed in class. Before submitting an assignment solution, please extract the jar file you intend to submit into a clean directory. Then compile and execute the program, from this jar file only. I.E., do what the grader will do.

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. You will loose points on 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 for 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 AIP and must turn in their own work. Any student found in violation of AIP may be given a failing grade for the course. Do not represent someone else's work as your own or 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 agreed to this policy.

AIP: [https://provost.asu.edu/academic-integrity](https://provost.asu.edu/academic-integrity)