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 credits)

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

1. Contact Information

Tim Lindquist (Tim.Lindquist@asu.edu)
Office Phone: 480 727 2783;
Class Web Site: http://quay.poly.asu.edu/Ser321
Zoom: https://zoom.us/j/5931499831

2. Office Hours

Office Hours: T,Th 1-3pm
Office Location: Peralta 230D

Contact the instructor via email should you be unable to make contact during office hours.

3. Course Topics, Course Objectives and Expected Learning Outcomes

There is significant C++ and Java software development required to complete Ser321. Also, the course is structured so you must use fundamental Linux program development tools for program development, testing, and 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 (C++) in the same semester. This is certainly do-able for a Software Engineering students, 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.

Topics Covered
a. Prerequisite review and introduction to course objectives.
b. Introduction to Linux and fundamental software development and software build tools, and deployment.
c. Serialization: Binary, Text-Based (XML & JSON), and custom.
d. Concurrency, threading, monitors, and synchronizing shared data.
e. Runtime environments, reflection, program execution paradigms, and use of pre-existing libraries (such as JSON and GUI).

f. Distribution, sockets, threaded servers, clients, and thread pooling.

g. Securing distributed applications: secure communications and securing the runtime environment.

h. Distribution paradigms including: Object-based, remote procedure call, client-server, peer-to-peer, service-oriented, and cloud. Protocols, such as http, dhcp, tcp/ip, and ftp.

i. Distributed object management: stateless, singleton and session.


Course Outcomes and their Relationship to Software Engineering Program Student Outcomes

a. Students are able to design and develop distributed applications that apply the concepts of concurrency, asynchronous behavior, threads, and shared data synchronization using software engineering best practices. Program Student Outcome supported: SER2.

b. Students have knowledge of fundamental program development tools available on Linux: (1) Command-line interpreter, (2) compiler, (3) debugger, and (4) build-tools to develop, test, and deploy distributed applications. Program Student Outcome supported: SER1.

c. Students are able to develop distributed applications consisting of multiple classes, multiple programs (such as client-server), and which utilize pre-existing libraries/frameworks (such as GUI, JSON, or JSON-RPC). Program Student Outcome supported: SER1.

d. Students have knowledge of serialization: (1) language built-in, (2) XML, and (3) JSON: Understand the role serialization plays in distributed applications; are able to utilize serialization in developing distributed applications. Program Student Outcome supported: SER1.

e. Students are able to apply common distribution paradigms appropriately for solving problems requiring distribution, such as client-server, service-oriented, and peer-to-peer. Program Student Outcome supported: SER1.

4. Grading Policies

Final grades are assigned in the course based on the typical scale of 90% and above is an A, 80% to less than 90% is a B, 70% to less than 80% is a C, and 60% to less that 70% is a D. Below 60% is failing. The instructor reserves the right to lower the cut-offs for letter grades based on student performance, but the cut-offs will not be higher than listed above. Plus-Minus grading may or may-not be used by the instructor. A student’s overall percentage in the course is calculated based on the weightings listed below. Note that Canvas may not reflect this weighting.

1. Laboratory assignments, and quizzes/worksheets are graded to account for 40% of the course grade.
2. A midterm exam accounts for 30% of the course grade, and
3. A final exam accounts for 30% of the course grade.

5. Absence, Make-Up, Assignment Late Submission, and Exam Scheduling Policies

Students are expected to adhere to the schedule posted for the course section as available on either or both of the class web site and/or Blackboard site for the course. This includes material coverage, assignment due dates, and examination times. Students who miss class for reasons other than those
described below will not be provided accommodation for exam rescheduling or late submissions. Late submission of programming assignments will receive from 0 to 50% of the total grades for the assignment, based on instructor discretion – primarily based on the criticality of the assignment to completing the remainder of course activities.

Accommodations will only be made for religious observances provided that students notify the instructor at the beginning of the semester concerning those dates. Students who expect to miss class due to official university-sanctioned activities should inform the instructor at the beginning of the semester. Alternative arrangements will be made for any examinations and other graded in-class work affected by religious observance and University-sanctioned activities only. The preceding policies are based on ACD 304–04, “Accommodation for Religious Practices” and ACD 304–02, “Missed Classes Due to University-Sanctioned Activities.”

6. **Readings, Assignments, Examinations, Special Materials, Required Activities**

Numerous textbook readings, instructor generated course notes, example software projects, constitute the content of this course and can be found on either the class web site, and/or the course Blackboard site. Unit-by-unit topical material is provided on the Class Schedule page of the course web site. These units are mapped to semester/session weeks by the course Blackboard site. Resources required to complete the course are explained on the Resources page of the course web site.

See the course required resources page at:
http://pooh.poly.asu.edu/Ser321/Resources/tools.html

See the Course Schedule and Detailed Topical Outline page at:
http://pooh.poly.asu.edu/Ser321/Schedule/schedule.html

7. **Classroom Behavior**

Cell phones and pagers must be turned off during class to avoid causing distractions. The use of recording devices is not permitted during class. Any violent or threatening conduct by anyone in this class will be reported to the ASU Police Department and the Office of the Dean of Students.

8. **Academic Integrity**

All students in this class are subject to ASU’s Academic Integrity Policy (available at http://provost.asu.edu/academicintegrity) and should acquaint themselves with its content and requirements, including a strict prohibition against plagiarism. All violations will be reported to the Dean’s office, who maintain records of all offenses. Students are expected to abide by the FSE Honor Code (http://engineering.asu.edu/integrity/).

Discussions among students on class material and laboratories are encouraged, but all students in this course must follow the AIP and must turn in their own work. That means that every source-code class making up a laboratory app that you submit in this class must have your individual copyright, and it must have been developed by you. In addition to claiming ownership of the code you deliver in this class, you must grant the Instructor and ASU the right to build, evaluate and demonstrate your code. Any student found in violation of this policy and/or ASU’s AIP may be given a failing grade for the course. Example ways you can violate ASU’s AIP is to represent someone else's work as your own (whether you
bought it or not) or if someone else represents your work as their own. Its fine to discuss concepts with others, and to help others locate and correct errors in their work. But, all graded work (exams, programming assignments, as well as any written exercises or quizzes) in this class must represent your own individual work. Grading may include executing student solutions using software that compares the structure and content of other student submissions, or other available solutions. Any cases of suspected violation of ASU’s AIP will be referred directly to the college office according to established policy. By your registration in this class, you are assumed to have read, understood and agreed to this policy.

ASU’s AIP: https://provost.asu.edu/academic-integrity

One ramification of this policy is that every student must assure that neither an electronic nor hard copy of their work is accessible to another student. If you share a computing system with someone else, you must know how to use access control to protect your files. If someone else steals your work (with or without your knowledge,) you may both get failing grade for the course.

9. **Disability Accommodations.**

Suitable accommodations will be made for students having disabilities and students should notify the instructor as early as possible if they will require same. Such students must be registered with the Disability Resource Center and provide documentation to that effect.

11. **Sexual Discrimination**

Title IX is a federal law that provides that no person be excluded on the basis of sex from participation in, be denied benefits of, or be subjected to discrimination under any education program or activity. Both Title IX and university policy make clear that sexual violence and harassment based on sex is prohibited. An individual who believes they have been subjected to sexual violence or harassed on the basis of sex can seek support, including counseling and academic support, from the university. If you or someone you know has been harassed on the basis of sex or sexually assaulted, you can find information and resources at https://sexualviolenceprevention.asu.edu/faqs.

As a mandated reporter, The instructor and support staff are obligated to report any information they may become aware of regarding alleged acts of sexual discrimination, including sexual violence and dating violence. ASU Counseling Services, https://eoss.asu.edu/counseling, is available if you wish discuss any concerns confidentially and privately.

**Notice:** Any information in this syllabus (other than grading and absence policies) may be subject to change with reasonable advance notice.

**Notice:** All contents of this course, its lectures, including written materials distributed to the class, published on Canvas or the class web site, and the example projects are all under copyright protection. Notes based on these materials may not be sold or commercialized without the express permission of the instructor, and adhering to any right-to-use license provided with the corresponding course artifact. [Note: Based on ACD 304-06.]