Course Syllabus

Note: Mostly set, but subject to change.

Course Schedule

Week	Activity	Video Length	Due Date / Window	Notes
1	OS Refresher (Optional)	2h 18m	Friday Jan 16	For students who need a refresher on AOS topics
1	Lesson 1: Intro to AOS	46m	Friday Jan 16	
2	Homework		Tuesday Jan 20 (11:59 PM ET)	1 week
2	Pre-lab		Tuesday Jan 20 (11:59 PM ET)	1 week
2	Lesson 2: OS Structures	2h 40m	Friday Jan 23	
3	Lesson 3: Virtualization	1h 53m	Friday Jan 30	
4–5	Lesson 4: Parallel Systems	5h 34m	Friday Feb 6 / Friday Feb 13	
6	Project 1: Virtual Machine Scheduling in KVM		Monday Feb 16 (11:59 PM ET)	3 weeks (available starting Monday Jan 26 at 11:59 PM ET)
6–7	Lesson 5: Distributed Systems	3h 20m	Friday Feb 20 / Friday Feb 27	
6–7	Test 1 (Lessons 1–4)		Friday Feb 20 (11:59 PM ET) – Monday Feb 23 (11:59 PM ET)	
8	Lesson 6: Distributed Objects and Middleware	1h 56m	Friday Mar 6	
9	Project 2: Barrier Synchronization		Monday Mar 9 (11:59 PM ET)	3 weeks (available starting Monday Feb 16 at 11:59 PM ET)
9–10	Lesson 7: Distributed Subsystems	3h 48m	Friday Mar 13 / Friday Mar 20	
	Spring Break		Monday Mar 23 - Friday Mar 27	Weekly Hangout will still occur
11	Project 3: Distributed Service using gRPC		Monday Mar 30 (11:59 PM ET)	3 weeks (available starting Monday Mar 9 at 11:59 PM ET)
11	Lesson 9: Internet Computing	2h 34m	Friday Apr 3	
12	Lesson 10: RT and Multimedia	1h 15m	Friday Apr 10	

Week	Activity	Video Length	Due Date / Window	Notes
12–13	Test 2 (Lessons 5–7)		Friday Apr 10 (11:59 PM ET) – Monday Apr 13 (11:59 PM ET)	
13	Lesson 8: Failures and Recovery	1h 58m	Friday Apr 17	
14	Lesson 11: Security	1h 17m	Friday Apr 24	
16	Project 4: Implement MapReduce Framework		Monday Apr 27 (11:59 PM ET)	4 weeks (available starting Monday Mar 30 at 11:59 PM ET)
17	Test 3 (Lessons 8–11)		Friday May 1 (11:59 PM ET) – Mon May 4 (11:59 PM ET)	

Reading List

Lecture	Papers
Lesson 2: OS Structures	1. Bershad et al., "Extensibility, Safety and Performance in the SPIN Operating System," SOSP 1995. 2. Engler, Kaashoek, O'Toole, "Exokernel: An Operating System Architecture for Application-Level Resource Management," SOSP 1995. 3. Liedtke, "On Microkernel Construction," SOSP 1995. 4. Liedtke, "Improved Address-Space Switching on Pentium Processors by Transparently Multiplexing User Address Spaces," GMD TR No. 933, Nov 1995 *(self-study)*.
Lesson 3: Virtualization	1. Barham, Dragovic, Fraser, Hand, Harris, Ho, Neugebauer, Pratt, Warfield, "Xen and the Art of Virtualization," SOSP 2003. 2. Waldspurger, "Memory Resource Management in VMware ESX Server," OSDI 2002.
Lesson 4: Parallel Systems	1. Mellor-Crummey & Scott, "Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors," TOCS, Feb 1991. 2. Bershad, Anderson, Lazowska, Levy, "Lightweight Remote Procedure Call," TOCS 8(1):37–55, Feb 1990. 3. *(partial reading: skip system modeling)* Squillante & Lazowska, "Using Processor-Cache Affinity Information in Shared Memory Multiprocessor Scheduling," IEEE TPDS, Feb 1993, 131–143. 4. Fedorova, Seltzer, Small, Nussbaum, "Performance of Multithreaded Chip Multiprocessors and Implications for Operating System Design," USENIX 2005. 5. Gamsa, Krieger, Appavoo, Stumm, "Tornado: Maximizing Locality and Concurrency in a Shared Memory Multiprocessor OS," OSDI 1999. 6. *(partial reading: Sec 1, 2, 3, 10)* Boyd-Wickizer et al., "Corey: An Operating System for Many Cores," OSDI 2008. 7. *(partial reading: Sec 1, 2, 3, 8)* Govil, Teodosiu, Huang, Rosenblum, "Cellular Disco: Resource Management Using Virtual Clusters on Shared-Memory Multiprocessors," SOSP 1999.

Lecture	Papers
Lesson 5: Distributed Systems	1. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System," CACM 21(7):558–565, July 1978. 2. Thekkath & Levy, "Limits to Low-Latency Communications on High-Speed Networks," TOCS, May 1993. 3. Hutchinson & Peterson, "The x-Kernel: An Architecture for Implementing Network Protocols," TSE 17(1):64–76, Jan 1991. 4. Wetherall, "Active Networks: Vision and Reality: Lessons from a Capsule-based System," SOSP 1999 (OSR 33(5)). 5. Liu, Kreitz, van Renesse, Hickey, Hayden, Birman, Constable, "Building Reliable High Performance Communication Systems from Components," SOSP 1999 (OSR 33(5)). 6. *(partial reading)* Schroeder & Burrows, "Performance of the Firefly RPC," SOSP 1989.
Lesson 6: Distributed Objects and Middleware	1. Mitchell et al., "An Overview of the Spring System," Compcon, Feb 1994. 2. Hamilton, Powell, Mitchell, "Subcontract: A Flexible Base for Distributed Programming," SOSP 1993. 3. Wollrath, Riggs, Waldo, "A Distributed Object Model for the Java System," USENIX COOTS, May 1996. 4. Cecchet, Marguerite, Zwaenepoel, "Performance and Scalability of EJB Applications," OOPSLA.
Lesson 7: Distributed Subsystems	1. Feeley, Morgan, Pighin, Karlin, Levy, Thekkath, "Implementing Global Memory Management in a Workstation Cluster," SOSP 1995. 2. Amza, Cox, Dwarkadas, Keleher, Lu, Rajamony, Yu, Zwaenepoel, "TreadMarks: Shared Memory Computing on Networks of Workstations," IEEE Computer, Feb 1996. 3. Anderson et al., "Serverless Network File System," *ACM Transactions on Computer Systems*, Feb 1996. 4. *(partial reading)* Satyanarayanan, "Coda: A Highly Available File System for a Distributed Workstation Environment," IEEE Trans. Computers, Apr 1990.
Lesson 8: Failures and Recovery	1. Satyanarayanan et al., "Lightweight Recoverable Virtual Memory," SOSP 1993, 146–160. 2. Lowell & Chen, "Free Transactions with Rio Vista," SOSP 1997. 3. Haskin et al., "Recovery Management in QuickSilver," TOCS, Feb 1988. 4. *(read on your own)* Gray et al., "The Recovery Manager of a Data Management System," ACM Computing Surveys 13(2), June 1981, 223–242. 5. *(partial reading: first 3 sections)* Porter, Hofmann, Rossbach, Benn, Witchel, "Operating System Transactions," SOSP 2009. 6. *(partial reading)* Peng & Dabek, "Large-scale Incremental Processing Using Distributed Transactions and Notifications," OSDI 2010.
Lesson 9: Internet Computing	1. Dean & Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters." 2. *(partial reading)* Brewer, "Lessons from Giant-Scale Services." 3. *(partial reading)* Barroso, Dean, Hölzle, "Web Search for a Planet: The Google Cluster Architecture," IEEE Micro. 4. Freedman, Freudenthal, Mazières, "Democratizing Content Publication with Coral." 5. DeCandia et al., "Dynamo: Amazon's Highly Available Key-value Store," SOSP 2007. 6. *(Read on your own — Web technologies)* 1. Curbera et al., "Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI," IEEE Internet Computing 6(2), 2002, 86–93.

Lecture	Papers
	2. Curbera, Khalaf, Mukhi, Tai, Weerawarana, "The Next Step in Web Services," CACM 46(10), Oct 2003, 29–34.
Lesson 10: RT and Multimedia	1. Goel, Abeni, Krasic, Snow, Walpole, "Supporting Time-Sensitive Applications on a Commodity OS," OSDI 2002. 2. Broomhead, Cremean, Ridoux, Veitch, "Virtualize Everything but Time," OSDI 2010. 3. Hilley & Ramachandran, "Persistent Temporal Streams," Middleware 2009. 4. Shahabi, Zimmermann, Fu, Yao, "Yima: A Second-Generation Continuous Media Server," IEEE Computer, June 2002.
Lesson 11: Security	1. Saltzer & Schroeder, "Protection and the Control of Information in Computer Systems," Proceedings of the IEEE 63(9):1278–1308, Sept 1975. 2. Satyanarayanan, "Integrating Security in Large-Scale Distributed Systems," TOCS, Aug 1989.

Note on partial readings: Partial readings without specified sections are not strictly required for exams and projects; however, if content is covered in lecture or the hangout, that content is fair game. If you choose a partial reading for a paper summary, you must read and summarize the entire paper.

Grade Distribution

- **Pre-lab:** 2%
- **Homework:** 3%
- **Projects:** 48%
  - Project 1: 12% *(individual)*
  - Project 2: 12%
  - Project 3: 12%
  - Project 4: 12%
- **Class Participation:** 5%
  - Ed discussion: 3%
  - Two paper summaries: 2%
- **Tests:** 42%
  - Test 1: 16%
  - Test 2: 14%
  - Test 3: 12%

Project partners: While project 1 must be completed individually, projects 2–4 may be done in pairs. You are responsible for finding a partner and managing logistics. The teaching team will not arbitrate partner issues (e.g., partner drops). We assume equal contribution and may randomly interview teams. You may also do projects solo. There is no special consideration for working alone.

Participation details: Provide answers to peer questions, ask questions, and collaborate on past exams. We will use forum summary stats (views, contributions, questions, answers). Only meaningful contributions will be considered; students should avoid spamming the forum with comments like "thank you", "great job", or "nice" unnecessarily.

Paper summaries: Sign up on the Wiki and choose two papers from the reading list to summarize (≈1 page each). If you choose a partial reading to summarize, read and summarize the entire paper. Sign-up link: **CS6210 OMSCS Spring 26 Paper Summary Sign Up**

(https://gtvault.sharepoint.com/:x:/s/Spring2026CS6210TAs/IQA6nZ3AdTlyQb59h5O3q0shASzKQzPxyz37lHxsucKJy8E?e=9Vl0Jt)

Exam Administration

- Tests are proctored with **Honorlock**.
- You may use one sheet of **blank** scratch paper. Show both sides to the webcam before starting.
- 80% of questions are released during the exam window. You may collaborate beforehand, but the exam itself is timed, closed-notes, individual, and proctored.

Extra Credit

- **Video Hangout attendance:** 0.5% for at least 10 appearances (attendance via PointSolutions quizzes; quizzes are ungraded and for attendance/comprehension).
- **Alternative (time-zone constraints):** 0.5% via an extra-credit assignment:
  - Summarize any 10 hangout recordings, ≤1 page each.
  - Submit as a single PDF.
  - Eligible only if you cannot attend hangouts and mark attendance via Turning Point.
- **CIOS bonus:** If final CIOS completion rate > 85%, everyone receives +0.5% to course total.

Course Policies

Honor Code

All students must know and follow the Georgia Tech Honor Code. See: **GT Student Honor Code** (https://osi.gatech.edu/students/honor-code) .

Not allowed:

- Copying, with or without modification, someone else's work when this work is not meant to be publicly accessible (e.g., a classmate's program or solution).
- Submission of material that is wholly or substantially identical to that created or published by another person or persons, without adequate credit notations indicating authorship (plagiarism)
- Putting your projects on public Github. Otherwise, if a student (in the future) copies your codes/projects, the student obviously violates the honor code but you will also be implicated.
- Copying code directly from online resources including ChatGPT is a violation of course policy.

You are encouraged to discuss problems and papers with others as long as this does not involve copying of code or solutions. Any public material that you use (open-source software, help from a text, or substantial help from a friend, etc...) should be acknowledged explicitly in anything you submit to us. If you have any doubt about whether something is legal or not please do check with the class Instructor or the TA.

Zero tolerance for any infraction. It pains me to say this but there have been incidents of infractions in previous semesters and the consequence is very serious and will affect your academic standing severely. So please do not put yourself (or me) in that situation.

Collaboration Policy

This class will consist of some individual assignments and some team projects. In order to facilitate a collaborative learning environment, you may discuss the projects and different approaches and techniques, as well as problems you may be having with each other. However, **all the code you turn in must be written by you**. For written homework, similarly, you may discuss the problems and concepts, but the actual answer that you submit must be written entirely by you. Please note that any infractions will be reported to the Office of Student Integrity (OSI) and could lead to suspension from the OMS program.

We will select random subsets of students to interview about their project submissions (since the class is too large to interview everybody about every project). In an interview, you will walk through your code with the TA and explain how it works. Therefore, it is important that you understand your own code and are able to explain it.

Late Policy

- Late submissions accepted for 24 hours after a deadline with a 20% penalty.
- **One Time Forgiveness (OTF):**
  - Each student has **one OTF** during the semester, usable on any graded work **except exams.**
  - An OTF allows submission up to **1 week past the posted deadline with no penalty**.
  - Late submissions are accepted for 24 hours after the OTF deadline (i.e., 8 days after the original deadline) with a 20% penalty.
  - For group projects: if one partner uses their OTF, the **extension applies to both partners**, but **only one partner** will be granted the extension in Gradescope; that partner must submit the project.
  - This means that if both partners each use their OTF on different projects, the pair could benefit from OTF twice in the semester.
  - To use OTF, complete the OTF request form here **CS 6210 OTF Form (https://forms.gle/oimQfmTobZMzzjkb9)** . Please check your spam or junk folder for the confirmation email.
  - If you face any issue, please create a private Ed post.