USC Viterbi
School of Engineering
Information Technology Program

TAC-483 "Communication for Software Engineers"
Units: 2
Fall
Tuesday/12:00pm – 1:50pm

Location: OHE 542

Instructor: Matthew Whiting

Office: RRB 221
Office Hours:
TBD see Piazza for latest
Contact Info:
whitingm@usc.edu

IT Help: Viterbi IT

Hours of Service:
Monday – Friday, 8:30 a.m. – 5:00 p.m.
Contact Info:
DRB 205
(213) 740-0517
engrhelp@usc.edu

Last updated: 3/30/25

Catalogue Description

Expanding beyond your current technical skills, user focused design, problem space vs solution space, data structures, algorithms, coding challenges, live engineering demonstrations and presentations.

Course Description

Communication and social skills are crucial to professional and advanced academic environments, and engineering students may not get enough practice in these areas. This class will help students prepare for the unique challenges of the transition from undergraduate study to the next phase of their careers. We will work specifically on the communication skills relevant to technical presentations in the context of both industry and academia such as technical interviews, research presentations, and other audiences.

The course is divided into two tracks designed to allow students to tailor their experience to best meet their career goals. The "Software Engineering" (SE) track will focus on algorithms and coding challenges, and the "Game Development" (GD) track will focus on the application of mathematics, physics, and geometry in the context of common video game scenarios.

Learning Objectives

By the end of this course, students will be able to:

1. Discuss complex algorithms using the technical vocabulary of computer science.
2. Adapt the discussion of complex computer science topics using non-technical language for discussions with product owners and other less technical team members.
3. Demonstrate the ability to think logically and creatively under pressure by solving technical challenges on the whiteboard in real-time in front of an audience.
4. Translate non-technical descriptions into algorithms and source code.
5. Apply algorithms in creative ways to solve abstract problems.

Prerequisite(s): CSCI-104 or TAC-365
Co-Requisite(s): n/a
Concurrent Enrollment: n/a
Recommended Preparation: Students should be familiar with or be willing to quickly learn Python, and it is presumed that students already have experience with data structures and common algorithms as covered in the prerequisite courses.

Course Notes

There are weekly lectures, and we will use Brightspace for course logistics. Slides during the course will be posted on Brightspace before the lecture begins.

A Piazza page will be used to communicate outside of class time.

Lecture topics will alternate between Third Space Thinking (TST) and software algorithms week-by-week.

TST represents the intersection between engineering and business and is characterized by communication and collaboration. The TST ideals help us to focus on understanding the needs of the client or customer so that we can design an engineering solution to effectively meet those needs.

Algorithm lectures will recap topics that students with the requisite knowledge have already learned while bringing a new focus on communication about these abstract topics in the context of different real-world situations such as the high-pressure environment of a technical interview or thesis defense.

With a significant focus on communication and social interaction, the course relies heavily on participation. Students will get practice in group activities and individual presentations.

To emulate the technical programming tests commonly included as part of job application, students will be asked to program algorithms which will be submitted and automatically graded using Github Classroom.

Finally, students will be called upon to expand their horizons on their own by finding industry-relevant articles and/or research papers. These will be presented by the students to practice valuable social communication skills that are not typically covered explicitly in software engineering courses.

Technological Proficiency and Hardware/Software Required

The live programming challenges can be conducted in the language of the student's choice, but graded programming homework assignments must be done in Python. Using Python will allow students to write clean and concise solutions, enabling them to focus on the algorithm and not the language's syntax. For a development environment, it is recommended that they install PyCharm to reduce code compatibility issues.

USC Technology Support Links
Zoom information for students
Blackboard help for students
Software available to USC Campus

Required Readings and Supplementary Materials

Cracking the Coding Interview: 189 Programming Questions and Solutions (6th Edition). Gayle Laakmann McDowell. ISBN-13: 978-0984782857.

Links for additional article readings are listed below.

Optional Materials

- *Mindset: The New Psychology of Success, Carol Dweck. ISBN-13 : 978-0345472328.*
- *The Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback, Dan Olsen. ISBN-13 : 978-1118960875.*
- *Managing Transitions: Making the Most of Change, William Bridges. ISBN-13 : 978-0738213804.*
- *Harvard Business Review on Negotiation and Conflict Resolution. ISBN-13: 978-1578512362.*
- *Mathematics for 3D game programming and computer graphics (3rd Edition).* Eric Lengyel. ISBN-13: 978-1-4354-5886-4.
- *Real-Time Collision Detection*. Christer Ericson. ISBN-13: 978-1-55860-732-3.
- *Essential Mathematics for Games and Interactive Applications (3rd Edition)*. James M. Van Verth, Lars M. Bishop. ISBN-13: 978-1482250923.

Description and Assessment of Assignments

Homework

In any case, homework needs to be completed individually prior to the beginning of the following class meeting. In addition, each student must be prepared to present their solutions for discussion with the class.

SE Track

Homework will be a combination of Third Space Thinking exercises and coding problems that apply to recently covered topics.

Technical homework assignments will be assigned every other week. These are python programs to be submitted on Github Classroom.

Other weeks include homework assignments which focus on the TST material. These are typically submitted via Brightspace.

GD Track

The homework assignments typically consist of (up to) 10 exercises worth 10 points each.

Each homework assignment concludes with a 10th exercise where the student is asked to demonstrate an understanding of the week's material and how it relates to the field of video game programming by creating their own problem along with the solution.

Assignments should be worked on paper and will be turned via Gradescope. Hand-written assignments are expected to show all work and include clearly understandable diagrams illustrating the concepts being applied.

Homework Presentation

At the beginning of each class, students will be expected to present their technical assignment solutions from the previous week. A few students will be selected, and each will explain their solution. This presents an opportunity for the presenter to practice with professional communication skills.

Each student is expected to present their technical homework assignment at least once during the semester.

In-Class Assignments

We will also have several in-class assignments to review the reading material for that week. These are usually group exercises and will be submitted on Brightspace.

Industry Article Write-Ups

On odd weeks, students will write a short summary (2-3 sentences) of a relevant tech article they must search out on their own. Articles must be taken from reputable tech news sites (for example techcrunch.com, wired.com), a major news source that has a tech section, or a research paper. Summaries are to be submitted on Brightspace, and at the beginning of class on even weeks, a few

students will be chosen to present their summary for class discussion. The write up will be graded based on completeness.

Each student is expected to present their industry article summary at least once during the semester.

Live Coding Challenges
On even weeks, several students will participate in a live assessment to demonstrate their knowledge and ability to articulate the covered topics. Each student will be asked to do a live coding challenge once during the semester. Students will be asked to write an algorithm on the whiteboard in front of the class. During the assessment, the other students will pay attention and make notes on what they thought was good/needed improvement. After the instructor's initial critique, the other students will have the opportunity to comment.

Midterm
The midterm exam will be a 20-minute oral exam where the students meet individually with the instructor to establish mastery of the material and presentation skills. The material in this exam will be cumulative to cover everything from the beginning of the semester up to this point.

There is no class meeting during that week. Instead, each student must schedule a 20-minute session with the instructor during that week.

Final Coding Challenge
In lieu of a final exam, students will be asked to complete a coding challenge on a predetermined course topic. The assessment can be completed from anywhere and will be administered via Github Classroom. The coding challenge will begin at the end of the final lecture meeting, and students will have until the end of the scheduled final period for the course to complete and submit it. This amounts to approximately 1 week depending on the final schedule for the semester in question.

Grading Breakdown

Assignment	**% of grade**
Homework	30
Homework Presentations	10
Industry Article Write-Ups	10
Live Coding Challenge	10
Midterm	20
Final Coding Challenge	20
TOTAL	100

Assignment Rubrics

Homework

SE Technical assignments are graded based on the ability to pass automated unit-tests. Each test passed is worth 10-20% of the overall score for that assignment and passing all tests will result in 100% for that assignment. These tests will cover the correctness of the algorithms, proper handling of edge-cases, and overall speed of the solution.

Most TST assignments are graded for correctness on Brightspace with equal weight given to each question.

The subjective TST assignments are graded for completeness.

The GD assignments are graded as follows: 40% of available points are given for any reasonable attempt, an additional 40% is earned for demonstration of a viable approach to solving the problem, and the final 20% will be awarded for a correct answer.

Homework Presentation

Homework presentation will be graded by participation. Each student is required to present a technical homework assignment at least once as well as an industry article at least once during the semester.

Industry Article Write-Ups

Industry article write-ups are graded for completeness and for meeting the standard of quality expected.

Live Coding Challenges

For the live coding challenges, 40% of the grade is based on the student's ability to apply their technical knowledge to solve the problem. 40% of the grade is based on the student's ability to use TST techniques to effectively communicate given time pressure. The final 20% of the grade is based on whether the right answer was achieved.

Final Coding Challenge

As with the Homework, the Final Coding Challenge will be graded using automated unit-tests on Github Classroom. These tests will cover the correctness of the algorithms, proper handling of edge-cases, and overall speed of the solution. Each test passed is worth 10-20% of the overall score for that assignment and passing all tests will result in 100% for that assignment.

Assignment Submission Policy

All homework assignments will be submitted via either Brightspace, Github Classroom, or Gradescope as appropriate. Assignments submitted via email will not be accepted.

All assignments are due prior to the beginning of the following class period.

Additional Policies

Assignments may be submitted late but will incur a 25% deduction for each day after the due date. Due to the focus on communication skills, attendance is mandatory despite there being no attendance grade.

In-class assessments and presentations cannot be rescheduled (except for an unforeseeable event) as this will disrupt the course schedule. If you know you will be missing any classes at the beginning of the semester, please tell the instructor as soon as possible.

Sharing of course materials outside of the learning environment

SCampus Section 11.12(B)

Distribution or use of notes or recordings based on university classes or lectures without the express permission of the instructor for purposes other than individual or group study is a violation of the USC Student Conduct Code. This includes, but is not limited to, providing materials for distribution by services publishing class notes. This restriction on unauthorized use also applies to all information, which had been distributed to students or in any way had been displayed for use in relationship to the class, whether obtained in class, via email, on the Internet or via any other media. (See Section C.1 Class Notes Policy).

Course Schedule: A Weekly Breakdown

Subject to change prior to the start of class on August 26, 2025

Notes: TST - Referenced from Third Space Thinking Syllabus. TST readings are listed by a number but listed at the bottom of this chart

There are 2 tracks. Each week will have a Homework assignment of some kind… you can choose either track each week.

Homework to be turned in on Brightspace, Github, or Gradescope as indicated by the color code.

Week	Topic	Homework Due SE	Homework Due GD	In-Class Assessments	Reading to do Before Session
1 Aug 26	Intro, Third Space Thinking Overview				McDowell §I-II, McDowell §IV-V
2 Sep 2	Big-O Algorithm Overview	Resume, Article write up	Resume, Article write up	Live Coding Assessment	McDowell §VI, TST [1]
3 Sep 9	TST: The importance of Mindset	Delete Item	Vectors		TST [2] TST [3] TST [4]
4 Sep 16	Strings, Sets, Dictionaries, Lists, Tuples	TBD, Article write up	Matrices, Article write up	Live Coding Assessment	McDowell §IX.1,
5 Sep 23	TST: Social Styles	String Permutations	Intersections		TST [5] TST [6] TST [7]
6 Sep 30	Queue, Stack, Linked List	Job Search, Article write up	Job Search, Article write up	Live Coding Assessment	McDowell §IX.2, McDowell §IX.3
7 Oct 7	TST: Problem Space Versus Solution Space	Stack Min	Triangles		TST [8] TST [9] TST [10]
8 Oct 14	Trees, Graphs	HW07, Article write up	Physics, Article write up	Live Coding Assessment	McDowell §IX.4, McDowell §VII
9 Oct 21	Midterm Exam Book a Time for an Oral Exam During the week				
10 Oct 28	TST: Adaptability - The New Competitive	Build Order	Algorithms		TST [11] TST [12] TST [13]

	Advantage				
11 Nov 4	Recursion & Dynamic Programming	HW09, Article write up	Application, Article write up	Live Coding Assessment	McDowell §IX.8, McDowell §VI
No Class Nov 11	No Class Veterans Day				
12 Nov 18	TST: How Not to Get a Job	Coins	C++		TST [14] TST [15] TST [16]
13 Nov 25	Sorting, Searching	HW11, Article write up	Fixed-Point, Article write up	Live Coding Assessment	McDowell §IX.9, McDowell §IX.10
14 Dec 2	TST: Statistics & Demographics	Sorted Search	Cache		TST [17] TST [18]
Didn't Fit	Code review, source control	Article write up	Article write up	Live Coding Assessment	McDowell §VIII
Didn't Fit	Course Wrap-Up		Group Activity: Mock Interviews		
Final		Final Coding Challenge	Due by the date of the scheduled final exam. Please see the Schedule of Classes for the University scheduled final exam date.		

List of TST Readings

Week 2

[1] How Emotional Intelligence Became a Key Leadership Skill – Harvard BuinessReview, Andrea Ovans
https://hbr.org/2015/04/how-emotional-intelligence-became-a-key-leadership-skill

Week 3

[2] The Importance of Mindset – whitepaper, Richard Miller
https://richard-k-miller.com/wp-content/uploads/2021/10/The-Importance-of-Mindset.pdf

[3] What Google Learned from its Quest to Build the Perfect Team
https://www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html

[4] Additional reading (optional):
Mindset: The New Psychology of Success, Chapters 1-3, Carol Dweck.

Week 5

[5] Agile versus Lean versus Design Thinking
https://medium.com/@jboogie/agile-vs-lean-vs-design-thinking-2329df8ab53c

[6] What Is Design Thinking and Why Is It So Popular?
https://www.interaction-design.org/literature/article/what-is-design-thinking-and-why-is-it-so-popular
[7] Additional reading (optional):
Olsen D (2015) The Lean Product Playbook, Introduction + Chapters 1 and 2

Week 7
[8] Problem Space vs Solution Space, Nikhil Gupta -
https://medium.com/@nikhilgupta08/problem-space-vs-solution-space-f970d4ace5c
[9] Why Human Centered Design Matters, Wired Magazine
http://www.wired.com/insights/2013/12/human-centered-design-matters/
[10] Additional reading (optional):
Olsen D (2015) The Lean Product Playbook, Chapters 3 and 4

Week 9
[11] Adaptability: The New Competitive Advantage, Harvard Business Review
https://hbr.org/2011/07/adaptability-the-new-competitive-advantage
[12] Constantly Changing Technologies: What's a Software Developer To Do?, Mashable
https://mashable.com/2010/12/20/constantly-changing-technologies/
[13] Additional reading (optional):
Bridges, W (2009) Managing Transitions, Chapter 6: How to Deal with Non-Stop Change

Week 11
[14] Ferrari, B (2012) The Executive's Guide to Better Listening, McKinsey Quarterly
https://www.mckinsey.com/featured-insights/leadership/the-executives-guide-to-better-listening#
[15] Mark Balbes (2014) Conflict and Resolution in the Agile World, ADTMag
https://adtmag.com/articles/2014/12/17/agile-conflict-resolution.aspx
[16] Additional reading (optional):
Harvard Business Review on Negotiation and Conflict Resolution, Chapter 3: The Team That Wasn't

Week 13
[17] Katzenbach, J (2012) Cultural Change that Sticks – Harvard Business Review -
https://hbr.org/2012/07/cultural-change-that-sticks
[18] Bennet, Milton J. (2014) The Development Model of Intercultural Sensitivity -
https://www.idrinstitute.org/dmis/

Statement on Academic Conduct and Support Systems

Academic Conduct:

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Part B, Section 11, "Behavior Violating University Standards" policy.usc.edu/scampus-part-b. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, policy.usc.edu/scientific-misconduct.

Support Systems:

Counseling and Mental Health - (213) 740-9355 – 24/7 on call
studenthealth.usc.edu/counseling
Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention.

National Suicide Prevention Lifeline - 1 (800) 273-8255 – 24/7 on call
suicidepreventionlifeline.org
Free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week.

Relationship and Sexual Violence Prevention Services (RSVP) - (213) 740-9355(WELL), press "0" after hours – 24/7 on call
studenthealth.usc.edu/sexual-assault
Free and confidential therapy services, workshops, and training for situations related to gender-based harm.

Office of Equity and Diversity (OED) - (213) 740-5086 | Title IX – (213) 821-8298
equity.usc.edu, titleix.usc.edu
Information about how to get help or help someone affected by harassment or discrimination, rights of protected classes, reporting options, and additional resources for students, faculty, staff, visitors, and applicants.

Reporting Incidents of Bias or Harassment - (213) 740-5086 or (213) 821-8298
usc-advocate.symplicity.com/care_report
Avenue to report incidents of bias, hate crimes, and microaggressions to the Office of Equity and Diversity |Title IX for appropriate investigation, supportive measures, and response.

The Office of Disability Services and Programs - (213) 740-0776
dsp.usc.edu
Support and accommodations for students with disabilities. Services include assistance in providing readers/notetakers/interpreters, special accommodations for test taking needs, assistance with architectural barriers, assistive technology, and support for individual needs.

USC Campus Support and Intervention - (213) 821-4710
campussupport.usc.edu
Assists students and families in resolving complex personal, financial, and academic issues adversely affecting their success as a student.

Diversity at USC - (213) 740-2101
diversity.usc.edu
Information on events, programs and training, the Provost's Diversity and Inclusion Council, Diversity Liaisons for each academic school, chronology, participation, and various resources for students.

USC Emergency - UPC: (213) 740-4321, HSC: (323) 442-1000 – 24/7 on call
dps.usc.edu, emergency.usc.edu
Emergency assistance and avenue to report a crime. Latest updates regarding safety, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible.

USC Department of Public Safety - UPC: (213) 740-6000, HSC: (323) 442-120 – 24/7 on call
dps.usc.edu
Non-emergency assistance or information.

Office of the Ombuds - (213) 821-9556 (UPC) / (323-442-0382 (HSC)
ombuds.usc.edu
A safe and confidential place to share your USC-related issues with a University Ombuds who will work with you to explore options or paths to manage your concern.