```
Field1,Field2,Field5_text,Field6_text,Field7,Field8_text
ES 2-01,"MWF 10:30-11:45
Anderson Hall 112",Introduction to Computing in
Engineering,E. Danahy,E. Danahy,
CS 4-01,,Teaching Computer Science,M. Monroe,M. Monroe,
CS 5-01,"T 4:30-6:30p
Eaton Hall 201",Preparing for Career Success,,K. Donoghue,
CS 10-01,"MW 1:30-2:45
Room To Be Announced",Computer Science for All,,E. Cornwall,
CS 11-M1,"T 5:30p-7:00p
Online (synchronous)",Introduction to Computer
Science,,P. Biswas,CS 11-M1
CS 11-01,"TR 10:30-11:45
Room To Be Announced",Introduction to Computer
Science,,E. Cornwall,CS 11-01
CS 11-02,"TR 12:00-1:15
Room To Be Announced",Introduction to Computer
Science,,E. Cornwall,CS 11-02
CS 14-01,,Emerging Scholars in Computer
Science,R. Townsend,R. Townsend,
CS 15-M1,"R 5:30p-7:00p
Online (synchronous)",Data Structures,,C. Magnano,CS 15-M1
CS 15-01,"MW 10:30-11:45
Room To Be Announced",Data Structures,,M. Kazerounian,CS 15-01
CS 15-02,"MW 3:00-4:15
Room To Be Announced",Data Structures,,M. Kazerounian,CS 15-02
CS 20-01,"TR 12:00-1:15
Room To Be Announced",Web Programming,,L. DiOrio,CS 20-01
CS 21-01,"TR 4:30-5:45p
Room To Be Announced",Concurrent Programming,M. Sheldon,M. Sheldon,
CS 23-01,"T 6:00p-9:00p
Room To Be Announced",Game Design,,J. Wiser,CS 23-01
CS 27-01,"TR 10:30-11:45
Room To Be Announced",How Systems Fail,,M. Kazerounian,
CS 28-01,"TR 1:30-2:45
Room To Be Announced",Cyber Security and Cyber
Warfare,M. Chow,"M. Chow,  J. Taliaferro",
CS 30-M1,"W 6:00p-7:30p
Online (synchronous)",Programming for Data Science,A. Couch,A. Couch,
CS 40-01,"TR 1:30-2:45
Room To Be Announced",Machine Structure & Assembly-Language
Programming,N. Mendelsohn,"N. Mendelsohn,  M. Sheldon",CS 40-01
CS 61-M1,"T 7:00p-8:30p
Online (synchronous)",Discrete Mathematics,,M. Jahn,CS 61-M1
CS 61-01,"MWF 9:30-10:20
Room To Be Announced",Discrete Mathematics,,M. Allen,CS 61-01
CS 61-02,"MWF 10:30-11:20
Room To Be Announced",Discrete Mathematics,,M. Allen,CS 61-02
CS 93-01,,Directed Study,,Members of the Department,
CS 98-01,"F 12:00-2:45
```

```
Room To Be Announced",Senior Capstone Project II,,D. Lillethun,
CS 99-01,,,M. Chow,M. Chow,
CS 105-01,"MW 3:00-4:15
Room To Be Announced",Programming
Languages,R. Townsend,R. Townsend,CS 105-01
CS 111-M1,"T 7:00p-8:30p
Online (synchronous)",Operating Systems,,P. Biswas,
CS 111-01,"TR 4:30-5:45p
Joyce Cummings Center 170",Operating Systems,,D. Lillethun,
CS 114-01,"TR 12:00-1:15
Room To Be Announced",Network Security,D. Votipka,D. Votipka,
CS 115-M1,"T 7:00p-8:30p
Online (synchronous)",Database Systems,,C. Doucette,CS 115-M1
CS 115-01,"M 7:00p-8:30p
Room To Be Announced",Database Systems,,C. Doucette,CS 115-01
CS 116-M1,"W 5:30p-7:00p
Online (synchronous)",Introduction to Security,M. Chow,M. Chow,CS 116-
M1
CS 116-01,"TR 4:30-5:45p
Room To Be Announced",Introduction to
Security,M. Chow,M. Chow,CS 116-01
CS 119-M1,"M 5:30p-7:00p
Online (synchronous)",Big Data,,J. Singh,CS 119-M1
CS 119-01,"MW 10:30-11:45
Room To Be Announced",Big Data,,J. Singh,CS 119-01
CS 120-M1,"T 5:30p-7:00p
Online (synchronous)",Web Programming and Engineering,,L. DiOrio,
CS 121-M1,"W 7:30p-9:00p
Online (synchronous)",Software Engineering,,Staff,CS 121-M1
CS 122-01,"MW 1:30-2:45
Room To Be Announced",Parallel Computing,,J. Grodstein,
CS 125-01,"MW 1:30-2:45
Joyce Cummings Center 265",Numerical Analysis,,S. Lee,
CS 131-M1,"M 7:30p-9:00p
Online (synchronous)",Artificial Intelligence,,F. Santini,CS 131-M1
CS 131-01,"MW 6:00p-7:15p
Room To Be Announced",Artificial Intelligence,,F. Santini,CS 131-01
CS 132-01,"MW 4:30-5:45p
Room To Be Announced",Computer Vision,,R. Shilkrot,
CS 135-M1,"F 5:30p-7:00p
Online (synchronous)",Introduction to Machine
Learning,,C. Magnano,CS 135-M1
CS 135-01,"TR 12:00-1:15
Room To Be Announced",Introduction to Machine
Learning,,M. Hughes,CS 135-01
CS 138-M1,"R 7:00p-8:30p
Online (synchronous)",Reinforcement Learning,,Y. Shukla,
CS 138-01,"MW 3:00-4:15
Room To Be Announced",Reinforcement Learning,J. Sinapov,J. Sinapov,
CS 142-01,"MW 4:30-5:45p
```

```
Room To Be Announced",Network Science,L. Cowen,L. Cowen,
CS 144-01,"T 1:30-4:00
Room To Be Announced",Iterative Methods in Machine
Learning,U. Khan,U. Khan,
CS 150-01,"TR 9:00-10:15
Room To Be Announced",HCI for Disability,E. Short,E. Short,
CS 150-02,"MW 10:30-11:45
Room To Be Announced",,R. Townsend,R. Townsend,
CS 150-04,"MW 10:30-11:45
Room To Be Announced",,,F. Dogar,
CS 150-05,"TF 12:00-1:15
Room To Be Announced",,R. Mirsky,R. Mirsky,
CS 150-06,,MITRE eCTF,M. Chow,"S. Bell,  M. Chow",
CS 150-07,"T 5:15p-8:15p
Room To Be Announced",Entrepreneurship for Computer Scientists,,,
CS 150-08,"TR 12:00-1:15
Room To Be Announced",Topics in Computational
Complexity,,V. Podolskii,
CS 151-01,"TR 10:30-11:45
Room To Be Announced","Addressing Cyber Threats,
Vulnerabilities",,L. Weissinger,
CS 151-02,"M 5:30p-7:00p
Joyce Cummings Center 402",Cybersecurity Clinic,M. Chow,M. Chow,
CS 151-03,"TR 4:30-5:45p
Room To Be Announced","Privacy, Security, and Data",,J. Bater,
CS 151-04,"TR 1:30-2:45
Room To Be Announced",Anonymous Communications Theory,,M. Ando,
CS 151-05,"MW 1:30-2:45
Room To Be Announced",Quantum Computer Science,,S. Mehraban,
CS 151-06,"MW 4:30-5:45p
Room To Be Announced",Debugging Cloud Computing,,R. Sambasivan,
CS 151-07,"TR 9:00-10:15
Room To Be Announced",Sustainable Computing,,H. Sundar,
CS 160-M1,"W 7:00p-9:00p
Online (synchronous)",Algorithms,A. Winslow,A. Winslow,CS 160-M1
CS 160-01,"MW 9:00-10:15
Room To Be Announced",Algorithms,,K. Edwards,CS 160-01
CS 167-01,"TR 10:30-11:45
Room To Be Announced",Computational
Biology,D. Slonim,D. Slonim,CS 167-01
CS 170-M1,"R 5:30p-7:00p
Online (synchronous)",Computation Theory,A. Winslow,A. Winslow,CS 170-
M1
CS 170-01,"TR 1:30-2:45
Room To Be Announced",Computation Theory,M. Monroe,M. Monroe,CS 170-01
CS 171-01,"MW 1:30-2:45
Room To Be Announced",Human-Computer
Interaction,R. Jacob,R. Jacob,CS 171-01
CS 178-01,"TR 1:30-2:45
Room To Be Announced",Visual Analytics,R. Chang,R. Chang,CS 178-01
```

```
CS 183-01,"MTWRF 12:00a-12:00a
Room To Be Announced",Privacy in the Digital Age,,,
CS 185-01,"R 6:30p-9:00p
Room To Be Announced",Computing for Developing Regions,,F. Dogar,
CS 191-01,,Research,,Members of the Department,
CS 193-01,,Directed Study,,Members of the Department,
CS 193-02,,Geometric Folding Algorithms,D. Souvaine,D. Souvaine,
CS 193-03,,MS CoreComp,,K. Edwards,
CS 197-01,,Honors Thesis,,Members of the Department,
CS 203-01,"TR 10:30-11:45
Room To Be Announced",How Systems Fail,,M. Kazerounian,
CS 239-01,"MW 9:00-10:15
Room To Be Announced","Ethics for AI, Robotics, and Human Robot
Interaction",V. Sarathy,V. Sarathy,
CS 260-1,"MW 3:00-4:15
Room To Be Announced",Advanced Algorithms,L. Cowen,L. Cowen,CS 260-1
CS 288-M1,"R 6:00p-7:00p
Online (synchronous)",Master of Science Capstone Project I,,M. Allen,
CS 289-M1,"R 7:00p-8:00p
Online (synchronous)",Master of Science Capstone Project II,,M. Allen,
CS 291-01,"seminar: R 3:00-4:15
Joyce Cummings Center 270",Seminar in Computer
Science,D. Souvaine,"J. Redmond,  D. Souvaine",
CS 291-02,"seminar: F 1:30-2:30
Robinson Hall 253",Seminar in Computer Science,J. Sinapov,J. Sinapov,
CS 293-01,,Graduate Special Topics / Master's
Project,,Members of the Department,
CS 296-01,,Master's Thesis,,Members of the Department,
CS 297-01,,Graduate Research,,Members of the Department,
CS 299-01,,Internship in Computer Science,M. Chow,M. Chow,
CS 401-PT,,,,Members of the Department,
CS 402-FT,,,,Members of the Department,
CS 404-01,,,,A. Abdulrazzaq,
CS 405-TA,,,,Members of the Department,
CS 406-RA,,,,Members of the Department,
CS 501-PT,,,,Members of the Department,
CS 502-FT,,,,Members of the Department,
DS 93-01,,Directed Study,,Members of the Department,
DS 98-01,"TR 1:30-2:45
Online (synchronous)",Senior Capstone Project in Data Science
II,A. Couch,A. Couch,
DS 143-01,"T 1:20-4:20
Bromfield-Pearson 005",Data Science for Sustainability,,D. Sunter,
DS 143-02,"M 6:00p-9:00p
Anderson Hall 108",Data Science for Sustainability,,D. Sunter,
DS 153-02,"MW 6:00p-7:15p
Eaton Hall 201",Computer Vision,,R. Shilkrot,DS 153-02
DS 153-03,"MW 10:30-11:45
Collaborative Learning and Innovation Complex 401",,,F. Dogar,
DS 153-04,"TF 12:00-1:15
```

```
Tisch Library 316",,R. Mirsky,R. Mirsky,
DS 193-01,,Directed Study,,Members of the Department,
DS 288-M1,"T 7:00p-9:00p
Online (synchronous)",Master of Science Capstone Project I,,M. Allen,
DS 289-M1,"T 8:00p-9:00p
Online (synchronous)",Master of Science Capstone Project II,,M. Allen,
DS 293-01,,Masters Project in Data Science,,Members of the Department,
DS 299-01,,Internship In Data Science,A. Couch,A. Couch,
DS 404-01,,,,A. Abdulrazzaq,
```

Text
"ES 2-01 Introduction to Computing in Engineering
E. Danahy
MWF 10:30-11:45, Anderson Hall 112

E+ Block

An introduction to engineering problem-solving with the aid of computational software. Scientific computing concepts will be introduced including number representation, arrays, structured programming techniques, and good coding practices. Basic numerical and data analysis methods will be introduced including numerical differentiation and integration, matrix operations, descriptive statistics, curve fitting, and optimization. Examples drawn from a variety of engineering disciplines will give students extensive practice in coding solutions and applying them to data.

Prerequisite: MATH 32 recommended"
"CS 4-01 Teaching Computer ScienceM. MonroeThis course will prepare undergraduates to function effectively and efficiently as undergraduate teaching assistants. Through this course, students will learn pedagogical techniques that match learner needs; discuss ethical and social concerns that UTAs face in the course of a semester; and problem solve together issues that arise as teaching assistants. This course is designed in a learner centered model requiring your active and engaged participation. Through your willingness to share your experiences and expertise and your collaboration with your fellow UTA we will together construct meaningful solutions to difficulty situations. Faculty from Computer Science will participate in some of the sessions as co-facilitators. Students will be expected to complete short readings; keep a reflective blog of your learning as a teacher; give a short final presentation on a topic of interest that you want to explore in more depth to help you in your TA class."
"CS 5-01 Preparing for Career Success
K. Donoghue
T 4:30-6:30p, Eaton Hall 201

This course is focused on helping Computer Science students prepare for finding roles in the technology industry. It covers topics such as strategic thinking about opportunities, skills assessment, marketing your skills, resume and marketing materials preparation, interview and practice, improving presentation skills, and career networking. Most class sessions will be virtual, with some class meetings in person.

Prerequisite: Students interested in registering for this class should fill out the form emailed to CS and DS students."
"CS 10-01 Computer Science for All
E. Cornwall
MW 1:30-2:45, Room To Be Announced

G+ Block

Computers are indispensable tools for research. This does not only hold for more technical fields such as physics or chemistry but also for the Humanities and the Social Sciences. While most students are competent users of standard software such as word processing or spreadsheets, the real power of the computer is unleashed when we are able to program it ourselves and make it do exactly what we want it to do.

This course is aimed at people who want to learn how to use computer science to solve basic information processing problems, such as analyzing text data and performing elementary statistics on them. It will cover elementary principles of computer science and will teach the student to independently write their own programs in the computer language Python.

This course is meant for people who have little or no previous experience in computer science. Therefore, in this course we do not assume that the students already know how to write computer programs. However, computer programming is a skill, and learning a new skill takes substantial amounts of effort and time. So the fact that this course is aimed at beginners does not mean that it is easy, or that it will involve less work than our other introduction courses, like e.g. COMP 11. On the contrary, it is very likely the case that this course will involves more effort than other introductory programming courses, if only because the fact that we do not assume any previous experience means that the road to our goal is going to be longer.

IMPORTANT NOTE: Passing this course does NOT fulfill the A&S Mathematics distribution requirement."
"CS 11-M1 Introduction to Computer Science
P. Biswas
T 5:30p-7:00p, Online (synchronous)

The study of computer science centers on two complementary aspects of the discipline. First, computer science is fundamentally concerned with the problem-solving methodologies it derives from its foundational fields: the design principles of engineering, mathematical theory, and scientific empirical study. Second, these methodologies are applied in the complex context of a modern day computing system. In this course we will address both of these important aspects. As a means for developing your design skills, we will discuss the fundamental features of a high level, general purpose programming language -- namely C++-- and learn how to use it as a tool for problem solving. We will also consider the performance of solutions, and how to apply both analytical and empirical assessment techniques. Finally, we will explore the Unix operating system as a context for problem solving. (Additional 2 hr weekly lab time scheduled at first class meeting.)

Recommendations: High school algebra. No prior programming experience is necessary."
"CS 11-01 Introduction to Computer Science
E. Cornwall
TR 10:30-11:45, Room To Be Announced

D+ Block

The study of computer science centers on two complementary aspects of the discipline. First, computer science is fundamentally concerned with the problem-solving methodologies it derives from its foundational fields: the design principles of engineering, mathematical theory, and scientific empirical study. Second, these methodologies are applied in the complex context of a modern day computing system. In this course we will address both of these important aspects. As a means for developing your design skills, we will discuss the fundamental features of a high level, general purpose programming language -- namely C++-- and learn how to use it as a tool for problem solving. We will also consider the performance of solutions, and how to apply both analytical and empirical assessment techniques. Finally, we will explore the Unix operating system as a context for problem solving. (Additional 2 hr weekly lab time scheduled at first class meeting.)

Recommendations: High school algebra. No prior programming experience is necessary."
"CS 11-02 Introduction to Computer Science
E. Cornwall
TR 12:00-1:15, Room To Be Announced

F+ Block

The study of computer science centers on two complementary aspects of the discipline. First, computer science is fundamentally concerned with the problem-solving methodologies it derives from its foundational fields: the design principles of engineering, mathematical theory, and scientific empirical study. Second, these methodologies are applied in the complex context of a modern day computing system. In this course we will address both of these important aspects. As a means for developing your design skills, we will discuss the fundamental features of a high level, general purpose programming language -- namely C++-- and learn how to use it as a tool for problem solving. We will also consider the performance of solutions, and how to apply both analytical and empirical assessment techniques. Finally, we will explore the Unix operating system as a context for problem solving. (Additional 2 hr weekly lab time scheduled at first class meeting.)

Recommendations: High school algebra. No prior programming experience is necessary."
"CS 14-01 Emerging Scholars in Computer Science
R. Townsend

Weekly, peer-led workshops exploring topics in computer science. Emphasis on the collaborative and problem-solving nature of computer science. No prior programming experience is necessary. Students must apply to enroll – see departmental website for details.

Prerequisite: Prerequisite: first year or sophomore standing
Corequisite: COMP 10 or COMP 11."
"CS 15-M1 Data Structures
C. Magnano
R 5:30p-7:00p, Online (synchronous)

A second course in computer science. Data structures and algorithms are studied through major programming projects in the C++ programming language. Topics include linked lists, trees, graphs, dynamic storage allocation, and recursion.

Prerequisite: COMP 11 or consent. This course and COMP 50-01 (COMP 50-PSS) may not both be taken for credit."
"CS 15-01 Data Structures
M. Kazerounian
MW 10:30-11:45, Room To Be Announced

E+ Block

A second course in computer science. Data structures and algorithms are studied through major programming projects in the C++ programming language. Topics include linked lists, trees, graphs, dynamic storage allocation, and recursion.

Prerequisite: COMP 11 or consent. This course and COMP 50-01 (COMP 50-PSS) may not both be taken for credit."
"CS 15-02 Data Structures
M. Kazerounian
MW 3:00-4:15, Room To Be Announced

I+ Block

A second course in computer science. Data structures and algorithms are studied through major programming projects in the C++ programming language. Topics include linked lists, trees, graphs, dynamic storage allocation, and recursion.

Prerequisite: COMP 11 or consent. This course and COMP 50-01 (COMP 50-

PSS) may not both be taken for credit."
"CS 20-01 Web Programming
L. DiOrio
TR 12:00-1:15, Room To Be Announced

F+ Block

An introduction to techniques, principles, and practices of writing computer programs for the World Wide Web. Server and browser capabilities and limits. Media types, handlers, and limitations. Web programming languages and techniques. Web security, privacy, and commerce. Lectures augmented with programming projects illustrating concepts and current practice.

Prerequisite: COMP 11; or COMP 10 and consent."
"CS 21-01 Concurrent Programming
M. Sheldon
TR 4:30-5:45p, Room To Be Announced

L+ Block

When we learn to program, we specify problem solutions as a single sequence of computations in a fixed, determined order. But the world isn't like that. Deer run into the woods, people talk on their phones, it rains. Nothing forces these things to happen one at a time, in a fixed order. They happen concurrently. We want to write concurrent programs, because we want to model the real world, because our computer systems actually have concurrent activities, and also to improve the performance or usability of our programs. The ubiquity of distributed applications and modern, multicore processors makes concurrent programming an essential skill.

This course explores different models of concurrent programming: students will gain competence in conventional shared-memory threads programming, and at least one natively concurrent programming model (actors or CSP). Time permitting, we may look at other models. We'll look at classic problems (like deadlock) and synchronization mechanisms (semaphores, locks, barriers).

Students will complete a substantial team programming project using these tools and techniques, and they will present their work to the class.

Prerequisite: CS 15, or graduate/postbac standing"
"CS 23-01 Game Design
J. Wiser
T 6:00p-9:00p, Room To Be Announced

11+ Block

Principles, design, and development of games. Game structure, engineering, physics, testing, 2D and 3D rendering, user interfaces, sound, and animation. Security of online games. Applications of Economics, Music, and Psychology in crafting games. Projects include writing game design documents, developing an interactive fiction game, and building a functional game in a team.

Prerequisite: Recommended: Comp 15."
"CS 27-01 How Systems Fail
M. Kazerounian
TR 10:30-11:45, Room To Be Announced

D+ Block

Failure of computer systems within the larger context of complex systems, including the power grid and aviation. Failures of algorithms and protocols, engineering and implementation, systems and applications, people and culture. Attacks, attack recovery, security, privacy, and attribution. Case studies of failures and attacks, including distributed denial of service, Meltdown, Spectre, and spear-phishing attacks.

Prerequisite: COMP 13 or consent of instructor."
"CS 28-01 Cyber Security and Cyber Warfare
M. Chow, J. Taliaferro
TR 1:30-2:45, Room To Be Announced

H+ Block

Interdisciplinary analysis of cybersecurity in the United States and other countries, intended to introduce engineering students to policymaking and intelligence aspects of cybersecurity and liberal arts students to the technical constraints of computer networks and software. Hands-on activities including packet analysis, exploiting a vulnerable system, password cracking, social engineering, reconnaissance, and malware analysis. Examination of state and non-state actors engaged in cyber-espionage, counterintelligence, deterrence, and offensive cyber operations. Guest speakers from private sector, civil liberties groups, and intelligence community.

Prerequisite: PS 61: Introduction to International Relations (for PS and IR majors) or COMP 15: Data Structures (for CS majors in A&S or SOE)"
"CS 30-M1 Programming for Data Science
A. Couch
W 6:00p-7:30p, Online (synchronous)

Fundamentals of programming for data-intensive science. Data structures and algorithms for data manipulation, cleaning, and

preparation. Design of data manipulation programs. Coding standards and practices. Use and creation of software libraries. Techniques for improving program performance. Examples drawn from data preparation and transformation, statistical data analysis, machine learning, deep learning, and deep data science including recommendation systems and trend analysis."
"CS 40-01 Machine Structure & Assembly-Language Programming
N. Mendelsohn, M. Sheldon
TR 1:30-2:45, Room To Be Announced

H+ Block

In COMP 40, you will learn about both high-level programming design principles and the low-level structure of computing machines. Design strategies will focus on modularity, abstraction, and separation of interface from implementation. The following topics on machine structure are covered: memory, caches, registers, machine arithmetic, and bitwise operations. We will also investigate the structure of assembly code, relocatable object code, binary machine code, and the translations between them. You will gain a deep understanding of all of these concepts via large-scale, realistic programming projects.

Mandatory lab will be held Fridays: sign up in SIS.

See https://engineering.tufts.edu/cs/current-students/undergraduate/high-demand-enrollment for the form required to get approval to enroll in this class.

Prerequisite: COMP 15."
"CS 61-M1 Discrete Mathematics
M. Jahn
T 7:00p-8:30p, Online (synchronous)

(Cross-listed as Mathematics 61.) Sets, relations and functions, logic and methods of proof, combinatorics, graphs and digraphs.

Prerequisite: Math 32 or Computer Science 11 or permission of instructor."
"CS 61-01 Discrete Mathematics
M. Allen
MWF 9:30-10:20, Room To Be Announced

(Cross-listed as Mathematics 61.) Sets, relations and functions, logic and methods of proof, combinatorics, graphs and digraphs.

Prerequisite: Math 32 or Computer Science 11 or permission of instructor."
"CS 61-02 Discrete Mathematics

M. Allen
MWF 10:30-11:20, Room To Be Announced

E Block

(Cross-listed as Mathematics 61.) Sets, relations and functions, logic and methods of proof, combinatorics, graphs and digraphs.

Prerequisite: Math 32 or Computer Science 11 or permission of instructor."
"CS 93-01 Directed Study
Members of the Department

Guided study of an approved topic. Credit as arranged.

Prerequisite: Consent."
"CS 98-01 Senior Capstone Project II
D. Lillethun
F 12:00-2:45, Room To Be Announced

Implementation and testing of the project designed in CS 97. Implementation tools, strategies, and platforms. Testing and debugging methodologies. Maintenance and release management. Legal, ethical, and social impacts of computing.

Prerequisite: CS 97"
"CS 105-01 Programming Languages
R. Townsend
MW 3:00-4:15, Room To Be Announced

I+ Block

Principles and application of computer programming languages. Emphasizes ideas and techniques most relevant to practitioners, but includes foundations crucial for intellectual rigor: abstract syntax, lambda calculus, type systems, dynamic semantics. Case studies, reinforced by programming exercises. Grounding sufficient to read professional literature.

Prerequisite: COMP 15 (Data Structures) and one semester of Discrete Mathematics (COMP/MATH 22 or 61)."
"CS 111-M1 Operating Systems
P. Biswas
T 7:00p-8:30p, Online (synchronous)

(Crosslisted as EE 128). Fundamental issues in operating system design. Concurrent processes: synchronization, sharing, deadlock, scheduling. Relevant hardware properties of uniprocessor and multiprocessor computer systems.

Prerequisite: Recommendations: COMP 15 and either CS 40 OR EE 14."
"CS 111-01 Operating Systems
D. Lillethun
TR 4:30-5:45p, Joyce Cummings Center 170

L+ Block

(Crosslisted as EE 128). Fundamental issues in operating system design. Concurrent processes: synchronization, sharing, deadlock, scheduling. Relevant hardware properties of uniprocessor and multiprocessor computer systems.

Prerequisite: Recommendations: COMP 15 and either CS 40 OR EE 14."
"CS 114-01 Network Security
D. Votipka
TR 12:00-1:15, Room To Be Announced

F+ Block

Vulnerabilities, attacks, and mitigations at all layers of the network stack. Public and private key cryptography, confidentiality and authentication protocols, botnets, firewalls, intrusion detection systems, and communication privacy and anonymity.

Prerequisite: Prerequisites: Computer Science 15 or graduate standing.

Recommendations: Computer Science 40."
"CS 115-M1 Database Systems
C. Doucette
T 7:00p-8:30p, Online (synchronous)

Fundamental concepts of database management systems. Topics include: data models (relational, object-oriented, and others); the SQL query language; implementation techniques of database management systems (storage and index structures, concurrency control, recovery, and query processing); management of unstructured and semistructured data; and scientific data collections.

Prerequisite: COMP 15."
"CS 115-01 Database Systems
C. Doucette
M 7:00p-8:30p, Room To Be Announced

Fundamental concepts of database management systems. Topics include: data models (relational, object-oriented, and others); the SQL query language; implementation techniques of database management systems (storage and index structures, concurrency control, recovery, and query processing); management of unstructured and semistructured data; and scientific data collections.

Prerequisite: COMP 15."
"CS 116-M1 Introduction to Security
M. Chow
W 5:30p-7:00p, Online (synchronous)

A systems perspective on host-based and network-based computer security. Current vulnerabilities and measures for protecting hosts and networks. Firewalls and intrusion detection systems. Principles illustrated through hands-on programming projects.

Prerequisite: Comp 15."
"CS 116-01 Introduction to Security
M. Chow
TR 4:30-5:45p, Room To Be Announced

L+ Block

A systems perspective on host-based and network-based computer security. Current vulnerabilities and measures for protecting hosts and networks. Firewalls and intrusion detection systems. Principles illustrated through hands-on programming projects.

Prerequisite: Comp 15."
"CS 119-M1 Big DataJ. SinghM 5:30p-7:00p, Online (synchronous)
""Big Data"" deals with techniques for collecting, processing, analyzing and acting on data at internet scale: unprecedented speed, scale, and complexity.

This course introduces the latest techniques and infrastructures developed for big data including parallel and distributed database systems, map-reduce infrastructures, scalable platforms for complex data types, stream processing systems, and cloud-based computing. The course content will be a blend of theory, algorithms and practical (hands on) work.
Prerequisite: A beginning course in databases, familiarity with Python, shell programming, Java, Scala, and SQL."
"CS 119-01 Big DataJ. SinghMW 10:30-11:45, Room To Be AnnouncedE+ Block   ""Big Data"" deals with techniques for collecting, processing, analyzing and acting on data at internet scale: unprecedented speed, scale, and complexity.

This course introduces the latest techniques and infrastructures developed for big data including parallel and distributed database systems, map-reduce infrastructures, scalable platforms for complex data types, stream processing systems, and cloud-based computing. The

course content will be a blend of theory, algorithms and practical (hands on) work.
Prerequisite: A beginning course in databases, familiarity with Python, shell programming, Java, Scala, and SQL."
"CS 120-M1 Web Programming and Engineering
L. DiOrio
T 5:30p-7:00p, Online (synchronous)

Web applications are complex systems that deliver a plethora of functionality to a large number of users, and also exhibit unique behaviors and demands in terms of performance, scalability, usability, and security. Web engineering is an emerging and multidisciplinary process that is used to create quality web applications. This course will discuss the limits of current web technologies, the similarities and differences between web and software engineering, design, information and service architectures, content management, and testing disciplines. Frameworks such as Rails, Spring, and Symfony will be emphasized and used. Projects will involve search, cloud computing, location-based services, and mobile web development.

Prerequisite: Comp 15 and Comp 20, or Consent of Instructor."
"CS 121-M1 Software Engineering
Staff
W 7:30p-9:00p, Online (synchronous)

Software engineering is an engineered discipline in which the aim is the production of software products, delivered on time and within a set budget, that satisfies the client’s needs. It covers all aspects of software production ranging from the early stage of product concept to design and implementation to post-delivery maintenance. This course covers the major concepts and techniques of software engineering including understanding system requirements, finding appropriate engineering compromises, effective methods of design, coding, and testing, team software development, and the application of engineering tools so that students can prepare for their future careers as software engineers. The course will combine a strong technical focus with a project providing the opportunity to obtain hands-on experiences on entire phases and workflow of the software process.

Prerequisite: COMP 40, graduate standing, or instructor consent."
"CS 122-01 Parallel Computing
J. Grodstein
MW 1:30-2:45, Room To Be Announced

G+ Block

(Cross-listed w/ EE 155) Programming modern parallel computer architectures, especially GPUs and multi-core CPUs. Rationale for modern multi-core CPUs. Challenges of multi-threaded programming. High-performance software taking advantage of hardware caches, cache coherency, memory systems and parallel computation.

Prerequisite: Recommendations: EE 126 or COMP 40."
"CS 125-01 Numerical Analysis
S. Lee
MW 1:30-2:45, Joyce Cummings Center 265

G+ Block

Analysis of algorithms involving computation with real numbers. Interpolation, methods for solving linear and nonlinear systems of equations, numerical integration, methods for ordinary differential equations.

Prerequisite: Recommendations: MATH 51 and programming ability in a language such as C, C++, Fortran, Pascal, or Matlab."
"CS 131-M1 Artificial Intelligence
F. Santini
M 7:30p-9:00p, Online (synchronous)

History, theory, and computational methods of artificial intelligence. Basic concepts include representation of knowledge and computational methods for reasoning. One or two application areas will be studied, to be selected from expert systems, robotics, computer vision, natural language understanding, and planning.

Prerequisite: Comp 15 and either COMP/MATH 22 or 61 or familiarity with both symbolic logic and basic probability theory."
"CS 131-01 Artificial Intelligence
F. Santini
MW 6:00p-7:15p, Room To Be Announced

M+ Block

History, theory, and computational methods of artificial intelligence. Basic concepts include representation of knowledge and computational methods for reasoning. One or two application areas will be studied, to be selected from expert systems, robotics, computer vision, natural language understanding, and planning.

Prerequisite: Comp 15 and either COMP/MATH 22 or 61 or familiarity with both symbolic logic and basic probability theory."
"CS 132-01 Computer Vision
R. Shilkrot
MW 4:30-5:45p, Room To Be Announced

K+ Block

Introduction to low and intermediate levels of classic and modern Computer Vision. How to design algorithms that process visual scenes to automatically extract information. Fundamental principles and important applications of computer vision, including image formation, processing, detection and matching features, image segmentation, and multiple views. Basics of machine learning and deep learning for computer vision.

Prerequisite: Recommendations: CS 160 and Math 165"
"CS 135-M1 Introduction to Machine Learning
C. Magnano
F 5:30p-7:00p, Online (synchronous)

An overview of methods whereby computers can learn from data or experience and make decisions accordingly. Topics include supervised learning, unsupervised learning, reinforcement learning, and knowledge extraction from large databases with applications to science, engineering, and medicine.

Prerequisite: Comp 15 and COMP/MATH 22 or 61 or consent of instructor. (Comp 160 is highly recommended)."
"CS 135-01 Introduction to Machine Learning
M. Hughes
TR 12:00-1:15, Room To Be Announced

F+ Block

An overview of methods whereby computers can learn from data or experience and make decisions accordingly. Topics include supervised learning, unsupervised learning, reinforcement learning, and knowledge extraction from large databases with applications to science, engineering, and medicine.

Prerequisite: Comp 15 and COMP/MATH 22 or 61 or consent of instructor. (Comp 160 is highly recommended)."
"CS 138-M1 Reinforcement Learning
Y. Shukla
R 7:00p-8:30p, Online (synchronous)

""Reinforcement learning problems involve learning what to do - how to map situations to actions - so as to maximize a numerical reward signal."" - Sutton and Barto (""Reinforcement Learning: An Introduction"", course textbook)

This course will focus on agents that much learn, plan, and act in complex, non-deterministic environments. We will cover the main theory and approaches of Reinforcement Learning (RL), along with common software libraries and packages used to implement and test RL algorithms. The course is a graduate seminar with assigned readings and discussions. The content of the course will be guided in part by the interests of the students. It will cover at least the first several chapters of the course textbook. Beyond that, we will move to more advanced and recent readings from the field (e.g., transfer learning and deep RL) with an aim towards focusing on the practical successes and challenges relating to reinforcement learning.

There will be a programming component to the course in the form of a few short assignments and a final projects.

Approved as a category 2 elective in Data Science (analysis and interfaces).

Prerequisite: Students are expected to be proficient programmers in at least one of the following languages: C++, Java, or Python. Prior coursework (or experience) in Artificial Intelligence and/or Machine Learning is highly recommended, but not required."
"CS 138-01 Reinforcement Learning
J. Sinapov
MW 3:00-4:15, Room To Be Announced

I+ Block

""Reinforcement learning problems involve learning what to do – how to map situations to actions – so as to maximize a numerical reward signal."" - Sutton and Barto (""Reinforcement Learning: An Introduction"", course textbook)

This course will focus on agents that much learn, plan, and act in complex, non-deterministic environments. We will cover the main theory and approaches of Reinforcement Learning (RL), along with common software libraries and packages used to implement and test RL algorithms. The course is a graduate seminar with assigned readings and discussions. The content of the course will be guided in part by the interests of the students. It will cover at least the first several chapters of the course textbook. Beyond that, we will move to more advanced and recent readings from the field (e.g., transfer learning and deep RL) with an aim towards focusing on the practical successes and challenges relating to reinforcement learning.

There will be a programming component to the course in the form of a few short assignments and a final projects.

Approved as a category 2 elective in Data Science (analysis and interfaces).

Prerequisite: Students are expected to be proficient programmers in at least one of the following languages: C++, Java, or Python. Prior

coursework (or experience) in Artificial Intelligence and/or Machine Learning is highly recommended, but not required."
"CS 142-01 Network Science
L. Cowen
MW 4:30-5:45p, Room To Be Announced

K+ Block

Mathematical foundations of the study of graphs and networks that arise as social, biological and Internet networks. Random graph models, community structure and inference problems, network dynamics, cascading. Example networks drawn from the application domains will be case studies as a companion to the general mathematical theory.

Approved as a category 2 elective in Data Science (analysis and interfaces).

Prerequisite: Recommendations: MATH 70 or 72 or CS 135 or 160"
"CS 144-01 Iterative Methods in Machine Learning
U. Khan
T 1:30-4:00, Room To Be Announced

6 Block

(Cross-listed as EE 143) Design and analysis of modern machine learning methods with emphasis on convex and nonconvex problems, and centralized, federated, and distributed computational architectures. Topics include convergence, complexities, contractions, fixed point theorems, and perturbation techniques; gradient descent and stochastic gradient descent in addition to accelerated methods including Polyak and Nesterov momentum, minibatching, and variance reduction. State of the practice methods will be covered including Adagrad, Autogard, Adam, sgdm with applications in image classification and document clustering.

Prerequisite: MATH 70 and CS 11"
"CS 150-01 HCI for Disability
E. Short
TR 9:00-10:15, Room To Be Announced

T+ Block

This is a graduate-level course for research-oriented students (including seniors and MS students who are considering research careers). Through readings, discussion, and a substantial course project we will explore the variety of ways that the human-computer interaction (HCI) research community has addressed the needs of disabled users, both from the perspective of assistive technology, which develops tech to support disabled people in their daily lives, and from the perspective of accessibility, which considers how disabled people can access computers and computing technologies. We will read and discuss academic papers from the HCI community (especially papers from ASSETS and CHI) in parallel with personal narratives from disabled people themselves and readings from the disability studies literature.

By the end of the course, students will:

be familiar with core themes of disability studies and the disability justice movement as they relate to HCI.
understand key considerations of HCI research with and for disabled people.
be familiar with a variety of papers in HCI relating to accessibility and assistive technology across a wide range of intended user populations.
learn to engage critically with the HCI literature from both a technical and disability studies perspective.
complete an HCI research project relating to accessibility or assistive technology.
The final grade will be based on performance on the project and project milestones, as well as discussion of readings, including leading discussion. Participation in most class discussions and substantial amounts of reading and writing are key components of this iteration of the class."
"CS 150-06 MITRE eCTF
S. Bell, M. Chow

MITRE's Capture the Flag challenge"
"CS 150-07 Entrepreneurship for Computer Scientists
T 5:15p-8:15p, Room To Be Announced

(Cross listed with ELS 194-02) 150 ECS is an introductory entrepreneurship course for Computer Science students. The course provides an overview of entrepreneurship, develops an entrepreneurial perspective, and provides a framework for learning the fundamentals of the essential elements of entrepreneurial ventures, specifically directed toward software-related industries and products. Students learn how to develop their technical ideas into potential business opportunities, and to explore their likelihood of becoming viable businesses. They learn how to do market research, to develop go-to-market strategies, value propositions and to differentiate their products or services from actual or potential competitors. The course consists of a balance of lectures, projects, case studies and interaction with entrepreneurs and computer scientists who participate in entrepreneurial organizations."
"CS 150-08 Topics in Computational Complexity
V. Podolskii
TR 12:00-1:15, Room To Be Announced

F+ Block

In this course we will study the computational complexity in concrete computational models. The main focus will be on various techniques to prove lower and upper bounds on the computational complexity in these models. We will mostly consider decision trees, communication complexity and Boolean circuit complexity. We will also briefly discuss proof complexity and algebraic computation models.

Prerequisite: CS 61 and a course on probability (MATH 165 or EE 104) are required to take this class. MATH 70 and CS 170 are recommended. CS 160 would be helpful."
"CS 151-01 Addressing Cyber Threats, VulnerabilitiesL. WeissingerTR 10:30-11:45, Room To Be AnnouncedD+ BlockComputer and information security can be understood as a discipline dealing with risk to computer and information systems and the data they process. This class will cover the analysis, assessment, understanding and management of risk and its components (threat, threat actor, vulnerability, impact, likelihood) from a technical perspective as well as a managerial one:

- Understanding the concepts of threat, threat actor, vulnerability, likelihood and impact in the context of risk.
- Key technical and non-technical threats, vulnerabilities and risks:

- Hardware vulnerabilities (e.g. Spectre & Meltdown) and mitigations
- Software vulnerabilities (OWASP Top 10) and mitigations
- Network and Architecture vulnerabilities and mitigations
- Threats, vulnerabilities, and risks related to physical factors, human factors, and human-computer interaction
- The real world: where resources are limited and all the above factors interact

- Risk in an organizational context
- Quantitative and qualitative methodologies (e.g. FAIR, threat modeling) to collect data on, and understand, risk and its components.
- How to decide what (and how) to fix or address vulnerabilities, threats, and risks
- How to manage threats, vulnerabilities, and risks. How to deal with risks, e.g. how to mitigate technical issues.Prerequisite: CS 201, CS 203, CS 15, or CS graduate standing"

"CS 151-02 Cybersecurity Clinic
M. Chow
M 5:30p-7:00p, Joyce Cummings Center 402

Students are placed in interdisciplinary teams to work on a cybersecurity project with non-profit organizations in the Medford, Somerville, Cambridge, and Greater Boston communities.

Prerequisite: Department consent required."

"CS 151-03 Privacy, Security, and Data
J. Bater
TR 4:30-5:45p, Room To Be Announced

L+ Block

Organizations today collect and analyze massive amounts of information for important decision-making applications. However, these large-scale analytics often compromise sensitive user information, such as medical or financial records. In this course, we will survey and apply state of the art techniques in privacy-preserving data science, such as differential privacy and secure multi-party computation, to learn how to build systems that provide useful results while still respecting user privacy. Students will be expected to read research papers, give in-class presentations, and complete a final project utilizing real-world data.

Prerequisite: CS 115-level knowledge; some experience with Python, C++, and basic probability."
"CS 151-04 Anonymous Communications Theory
M. Ando
TR 1:30-2:45, Room To Be Announced

H+ Block

We know how to communicate a message so that only the recipient can read the message. (We can just encrypt the message.) But how can we communicate over the Internet without anyone learning *who* we are communicating with? Anonymous channels can help the Iranian protester who wishes to inform the world what is happening in the streets of Tehran by tweeting videos and the netizen in Moscow who wants to read the BBC news (currently banned in Russia). We could use Tor (i.e., “The onion router,” inspired by Chaum's onion routing idea) or VPN, but both are easily blocked, and neither guarantees privacy from the adversary with full view of the network traffic (e.g., a standard model for a resourceful ISP- or AS-level adversary). In this course, we present cryptography-style definitions of anonymity and study state-of-the-art techniques for achieving provable anonymity.

Prerequisite: Prerequisites: Math 21 and CS 170 or graduate standing."
"CS 151-05 Quantum Computer Science
S. Mehraban
MW 1:30-2:45, Room To Be Announced

G+ Block

The universe at the sub-atomic scale is governed by quantum mechanical laws, which fundamentally differ from classical laws of motion. What is the nature of computation in such scales? Can we use quantum mechanical particles to perform computations? These are the core questions of the field of quantum computing. In this course, we present an elementary-level introduction to the computer science foundations of quantum computing. Topics include Hilbert spaces, quantum entanglement, quantum measurements, quantum circuits, quantum protocols and algorithms, Hamiltonians and the ground state problem, and quantum error-correcting codes. Students from different areas of engineering and sciences, such as computer science, physics, electrical engineering, mathematics, or chemistry, who wish to learn about the computer science foundations of quantum computers can benefit from this class. The main focus of this course is on the theoretical foundations of quantum computing; mathematical enthusiasm and knowledge in areas such as linear algebra, algorithms, discrete mathematics, and calculus are required.

Prerequisite: Math 34 and Math 70 and one of (CS 61 or Math 61 or Math 65)"
"CS 151-06 Debugging Cloud Computing
R. Sambasivan
MW 4:30-5:45p, Room To Be Announced

K+ Block

Cloud computing, which is the practice of renting software and hardware services from providers who run large-scale data centers, has become critical to modern society. We rely on software running within cloud data centers when shopping (e.g., at Amazon), when conducting financial transactions (e.g., at an online broker), when collaborating at work (e.g., using Google Docs), and even when playing games (e.g., Fortnite). Failures or performance problems within these data centers or the software running on them can have widespread effects and be devastating.

In this course, we will examine failures in cloud environments and discuss important research on tools that use systems knowledge, machine learning, and statistics to help engineers diagnose them. To provide students with necessary background, we will start with a brief introduction to cloud computing and the software systems that make cloud computing possible. The course will involve reading research papers, homework assignments, and coding-based projects. It is recommended for graduate students and advanced upper-level undergraduates.

Prerequisite: CS 15 and CS 40 or graduate standing required; CS 111 recommended."
"CS 151-07 Sustainable Computing
H. Sundar
TR 9:00-10:15, Room To Be Announced

T+ Block

Data centers require enormous amounts of electricity to operate and cool, so understanding sustainable computing practices for improving energy efficiency can significantly reduce environmental impact. This course delves into cutting-edge research in system-level power, energy, and thermal management. We will explore techniques at the operating system, network, and application levels to optimize power consumption and thermal efficiency while maintaining performance and reliability.

Prerequisite: CS 40 or Graduate Standing"
"CS 160-M1 Algorithms
A. Winslow
W 7:00p-9:00p, Online (synchronous)

Introduction to the study of algorithms. Strategies such as divide-and-conquer, greedy methods, and dynamic programming. Graph algorithms, sorting, searching, integer arithmetic, hashing, and NP-complete problems.

Prerequisite: COMP 15 and COMP/MATH 22 or 61."
"CS 160-01 Algorithms
K. Edwards
MW 9:00-10:15, Room To Be Announced

R+ Block

Introduction to the study of algorithms. Strategies such as divide-and-conquer, greedy methods, and dynamic programming. Graph algorithms, sorting, searching, integer arithmetic, hashing, and NP-complete problems.

Prerequisite: COMP 15 and COMP/MATH 22 or 61."
"CS 167-01 Computational Biology
D. Slonim
TR 10:30-11:45, Room To Be Announced

D+ Block

Computational challenges in molecular biology, including sequence alignment and comparison, genomic annotation, micro array data analysis, and proteomics. Underlying computational techniques such as dynamic programming, hidden Markov models, statistical analyses, and search and optimization procedures. Prerequisites: Comp15 and at least one CS course numbered 100 or higher.

Prerequisite: Comp15 and at least one CS course numbered 100 or higher."
"CS 170-M1 Computation Theory
A. Winslow

R 5:30p–7:00p, Online (synchronous)

Models of computation: Turing machines, pushdown automata, and finite automata. Grammars and formal languages including context-free languages and regular sets. Important problems including the halting problem and language equivalence theorems.

Prerequisite: COMP 15 and COMP/MATH 22 or 61."
"CS 170-01 Computation Theory
M. Monroe
TR 1:30–2:45, Room To Be Announced

H+ Block

Models of computation: Turing machines, pushdown automata, and finite automata. Grammars and formal languages including context-free languages and regular sets. Important problems including the halting problem and language equivalence theorems.

Prerequisite: COMP 15 and COMP/MATH 22 or 61."
"CS 171-01 Human-Computer Interaction
R. Jacob
MW 1:30–2:45, Room To Be Announced

G+ Block

Introduction to human-computer interaction, or how computers communicate with people. Methodology for designing and testing user interfaces, interaction styles (command line, menus, graphical user interfaces, virtual reality), interaction techniques (including use of voice, gesture, eye movement), design guidelines, and user interface management software system. Students will design a small user interface, program a prototype, and test the result for usability.

Prerequisite: COMP 15"
"CS 178-01 Visual Analytics
R. Chang
TR 1:30–2:45, Room To Be Announced

H+ Block

Visual analytics is the science of combining interactive visual interfaces with automatic data science, machine learning, and AI algorithms to support analytical reasoning. Modern visual analytics tools help users synthesize information and derive insight from large, dynamic, ambiguous, and often conflicting data, and to communicate their findings effectively for decision-making. This course will serve as an introduction to the topic of visual analytics that will include lectures on both theoretical foundations and application

methodologies. The goals of this course are for students to: (1) learn about using visual analytics tools (e.g. Tableau), (2) become proficient in generating visualizations within popular data science tools (e.g. R, Python and scikit-learn), (3) develop their own visual analytics tools (in Javascript and D3), and (4) design evaluation methods to assess the effectiveness of these tools."
"CS 183-01 Privacy in the Digital Age
MTWRF 12:00a-12:00a, Room To Be Announced

This course will provide an introduction to the legal and regulatory protections for personal data and the evolving nature of digital surveillance and online privacy. The class will cover public and private sector threats to privacy and look at how different countries have implemented both privacy protections and surveillance regimes that affect individual Internet users and govern the collection of their personal data. Topics to be covered include encryption policy, law enforcement access to data, intelligence agency access to data, the European General Data Protection Regulation, the Electronic Communications Privacy Act, domestic and international lawful interception of data, protections for geolocation data and other forms of metadata, and facial recognition technology. This course will primarily focus on law and policy measures related to privacy but it will also cover some basic technical material related to cryptography and networking that is relevant to understanding the impacts of different policies."
"CS 185-01 Computing for Developing Regions
F. Dogar
R 6:30p-9:00p, Room To Be Announced

13 Block

An interdisciplinary approach to the role of computing technologies in developing regions. Low-cost communication infrastructure; socially relevant technologies for education, healthcare, and governance; and the use of technology by underserved communities and populations in developing regions. Problems and existing solutions covered through case studies. Group projects on designing, implementing, and evaluating a solution. Recommendations: CS10 or CS11 or some background in computer science and/or technology.

Prerequisite: Completion of COMP 15 or graduate standing"
"CS 191-01 Research
Members of the Department

Research on a topic in Computer Science or a related discipline, culminating in a final paper describing accomplishments, with the goal of advancing the state of the art. Topic is proposed by a faculty sponsor in Computer Science. Faculty consent required. Students sign up for a section that corresponds to a faculty member.

Prerequisite: Consent"
"CS 193-01 Directed Study
Members of the Department

Guided study of an approved topic. Credit as arranged.

Prerequisite: Consent."
"CS 193-02 Geometric Folding Algorithms
D. Souvaine

TBD

Prerequisite: Consent of department"
"CS 193-03 MS CoreComp
K. Edwards

TBD

Prerequisite: Consent of department"
"CS 197-01 Honors Thesis
Members of the Department

Honors Thesis Computer Science."
"CS 203-01 How Systems Fail
M. Kazerounian
TR 10:30-11:45, Room To Be Announced

D+ Block

Failure of computer systems within the larger context of complex systems, including the power grid and aviation. Failures of algorithms and protocols, engineering and implementation, systems and applications, people and culture. Attacks, attack recovery, security, privacy, and attribution. Case studies of failures and attacks, including distributed denial of service, Meltdown, Spectre, and spear-phishing attacks. A recitation and graduate-level assignments are required. Four credit hours.

Prerequisite: Graduate standing in a discipline other than Computer Science, Data Science, Bioinformatics, Cognitive Science, or Human-Robot Interaction."
"CS 239-01 Ethics for AI, Robotics, and Human Robot Interaction
V. Sarathy
MW 9:00-10:15, Room To Be Announced

R+ Block

Technical challenges of endowing autonomous artificial agents with normative principles that will allow them to operate successfully in human societies. Algorithmic approaches in artificial agents (rule-based, utility-based, behavior-based, etc.) and their links to philosophical foundations of the main ethical theories (virtue ethics, deontology, utilitarianism). Conceptual and mathematical analysis of assumptions underlying each algorithmic approach as well as implementation in agent-based simulations. Functional and performance tradeoffs of the algorithms and their implications for autonomous robots and AI systems. Mathematical and computational challenges of the different proposals for ""implicit"" and ""explicit"" ethical agents, including inverse reinforcement learning, verification-based approaches, and model checking.

Prerequisite: CS MS or PhD or DS MS or HRI MS"
"CS 260-1 Advanced AlgorithmsL. CowenMW 3:00-4:15, Room To Be AnnouncedI+ BlockIf you loved your algorithms class and can't wait for more, this is the class for you. In this pleasant and fun class, we will look at some more modern algorithms, some beautiful algorithms gems, and some areas of current research in algorithms. Topics will include using randomness in the design and analysis of algorithms, approximation algorithms, and online algorithms.Prerequisite: Comp 160 or permission of the instructor."
"CS 288-M1 Master of Science Capstone Project I
M. Allen
R 6:00p-7:00p, Online (synchronous)

Part one of a two-course, hands-on, and project-based culmination to the Master of Science in Computer Science Online program. Application of principles, strategies, methods, and tools for requirements analysis and design of a programming project, including project planning, project management, and proof of concept prototyping. Formulation of a project plan, including estimation of project completion requirements and timeline. To be taken in the second-to-last term of the Master of Science in Computer Science Online degree. Not available to students outside that program.

Prerequisite: CS 180 or 121, and enrollment in the Master of Science in Computer Science Online program."
"CS 289-M1 Master of Science Capstone Project II
M. Allen
R 7:00p-8:00p, Online (synchronous)

Part two of a two-course, hands-on, project-based culmination experience for the Master of Science in Computer Science Online program. Implementation of the project defined in part one, including use of principles, tools, and strategies for implementation, debugging, testing, documentation, maintenance, and release management. Presentation of final project results and documentation. To be taken in the last term of the Master of Science in Computer

Science Online degree. Not available to students outside that program.

Prerequisite: CS 288, and enrollment in the Master of Science in Computer Science Online program."
"CS 291-01 Seminar in Computer Science
J. Redmond, D. Souvaine
seminar: R 3:00-4:15, Joyce Cummings Center 270

J+ Block

A weekly seminar with guest speakers discussing research challenges and recent advances in Computer Science. Pass/fail only. To receive credit, students must attend and provide feedback for at least 50% of the seminars.

Prerequisite: Ph.D. standing in Computer Science"
"CS 291-02 Seminar in Computer Science
J. Sinapov
seminar: F 1:30-2:30, Robinson Hall 253

A weekly seminar with guest speakers discussing research challenges and recent advances in Computer Science. Pass/fail only. To receive credit, students must attend and provide feedback for at least 50% of the seminars.

Prerequisite: Ph.D. standing in Computer Science"
"CS 293-01 Graduate Special Topics / Master's Project
Members of the Department

Guided individual study of an approved topic suitable for a master's design project. Credit to be arranged. Members of the Department.

Prerequisite: Permission of the instructor."
"CS 296-01 Master's Thesis
Members of the Department

Guided research on a topic which has been approved as a suitable subject for a master's thesis. Credit as arranged. Supervision and advising by faculty of the Department. Please see the Registrar for appropriate section numbers."
"CS 297-01 Graduate Research
Members of the Department

Guided research on a topic suitable for a doctoral dissertation."
"CS 299-01 Internship in Computer Science
M. Chow

Study of approved topics in Computer Science in concert with an internship in computing or a related field outside the University.

Prerequisite: Permission of instructor"
"DS 93-01 Directed Study
Members of the Department

Directed study, as approved by professor and department."
"DS 98-01 Senior Capstone Project in Data Science II
A. Couch
TR 1:30-2:45, Online (synchronous)

H+ Block

A continuation of COMP 87. Analysis of the problem proposed in COMP 87 is completed and a final paper summarizes data gathered, analytic results, lessons learned, and opportunities for future study.

Prerequisite: COMP 87"
"DS 143-01 Data Science for Sustainability
D. Sunter
T 1:20-4:20, Bromfield-Pearson 005

6+ Block

Crosslisted as ME 193. This course explores emerging topics in data science and statistical learning with applications to the three pillars of sustainability (environmental, economic, and social). Students learn to build, estimate, and interpret models that describe phenomena in the broad area of energy and environmental decision-making with an emphasis on social justice. Students leave the course as both critical consumers and responsible producers of data-driven analysis. The objectives of this class include i) learning a suite of data-driven modeling and prediction tools, ii) building the programming and computing expertise to use those tools, and iii) developing the ability to formulate an analysis to answer sustainability questions of interest to industry and/or government partners.

Prerequisite: This course uses Python. Prior experience with statistics and programming is required."
"DS 143-02 Data Science for Sustainability
D. Sunter
M 6:00p-9:00p, Anderson Hall 108

10+ Block

Crosslisted as ME 193. This course explores emerging topics in data science and statistical learning with applications to the three pillars of sustainability (environmental, economic, and social). Students learn to build, estimate, and interpret models that describe phenomena in the broad area of energy and environmental decision-making with an emphasis on social justice. Students leave the course as both critical consumers and responsible producers of data-driven analysis. The objectives of this class include i) learning a suite of data-driven modeling and prediction tools, ii) building the programming and computing expertise to use those tools, and iii) developing the ability to formulate an analysis to answer sustainability questions of interest to industry and/or government partners.

Prerequisite: This course uses Python. Prior experience with statistics and programming is required."
"DS 153-02 Computer Vision
R. Shilkrot
MW 6:00p-7:15p, Eaton Hall 201

M+ Block

This course is an introduction to low and intermediate level Computer Vision. We will learn how to design algorithms that process visual scenes to automatically extract information. The course will cover fundamental principles and important applications of computer vision, including image formation, processing, detection and matching features, image segmentation, and multiple views."
"DS 193-01 Directed Study
Members of the Department

Directed study, as approved by professor and department."
"DS 288-M1 Master of Science Capstone Project I
M. Allen
T 7:00p-9:00p, Online (synchronous)

Part one of a two-course, hands-on, and project-based culmination to the Master of Science in Data Science Online program. Application of principles, strategies, methods, and tools of Data Science to analyze data and justify real-world decisions based upon those data. The first term includes creation of a proof of concept prototype of a chosen data analysis project. To be taken in the second-to-last term of the Master of Science in Data Science Online degree."
"DS 289-M1 Master of Science Capstone Project II
M. Allen
T 8:00p-9:00p, Online (synchronous)

Part two of a two-course, hands-on, and project-based culmination to the Master of Science in Data Science Online program. A continuation of the work started in DS 288, culminating in a complete analysis of the problem chosen in DS 288 as well as a final paper describing the analytic process and decisions and/or recommendations indicated by the data. To be taken in the last term of the Master of Science in Data

Science Online degree."
"DS 293-01 Masters Project in Data Science
Members of the Department

Guided individual study of an approved topic suitable for a master's project in Data Science."
"DS 299-01 Internship In Data Science
A. Couch

Study of approved topics in Computer Science in concert with an internship in computing or a related field outside the University.

Prerequisite: Permission of instructor"

MSCS & PhD Core Competencies

Students are required to demonstrate competency in four core areas.

This can be done through taking one course in each area while at Tufts.

Or, if you have previously taken courses in these areas, you might be excused from repeating the requirement at Tufts.

Program Languages (PL)

- CS 21: Concurrent Programming (No Graduate Credit)
- CS 86: Object-Oriented Programming for GUIs (No Graduate Credit)
- CS 105: Programming Languages
- CS 121: Software Engineering
- CS 107: Compilers

Computer Architecture & Assembly Language (CA&AL)

- CS 40: Machine Structure (No Graduate Credit)
- CS 111: Operating Systems
- CS 112: Networks
- CS 114: Network Security
- CS 116: Introduction to Security
- CS 118: Cloud Computing
- CS 146 / EE 126: Computer Engineering
- CS 107: Compilers
- CS 140: Advanced Topics in Computer Architecture

Theory of Computation (ToC)

- COMP/CS 61: Discrete Math
- CS 170: Computation Theory

Data Structure & Analysis of Algorithms (DS&AA)

- CS 160: Intro to Algorithms

Registration

Continuous Enrollment Policy

Graduate students must be enrolled (registered), or on an approved leave of absence, for every academic-year semester between matriculation and graduation. Graduate students may only register for courses that count toward their degree program. Students should register during the early registration periods in November and April. The university reserves the right to withhold registration for any student having unmet Academic Integrity Training (AIT), financial, or health services obligations. Students who fail to register by the end of the first week of classes, or the add deadline for the term, whichever is first, will be administratively withdrawn and subject to a $350 reinstatement fee. International students must maintain proper enrollment status per their visa requirements.

Degree-only Status

If a student has registered for all required courses, including thesis research (295, 296) or dissertation research (297, 298), the student must register for a course designation that indicates that only thesis, project, master's exhibition, or dissertation-related work, whether part-time or full-time, is being pursued.

Master's candidates must register for course 401-PT (part-time) or 402-FT (full-time) in their department and doctoral candidates must register for course 501-PT (part-time) or 502-FT (full-time) in their department. Graduate students may declare full-time status of thirty-five hours per week (402 or 502) with their advisor's concurrence.

Enrollment Status

Graduate students are responsible for maintaining enrollment status at Tufts.* Federal regulations require students to be enrolled (registered) full-time or half-time in order to receive and/or defer student loans. Tufts provides information regarding student enrollment status to lenders via the National Student Loan Clearinghouse and is required by law to return funds for students who do not maintain a minimum of half-time enrollment status. Enrollment status is either full-time, half-time, or part-time as defined below:

Full-time: Nine (9) or more course credits, six (6) course credits and a teaching or research assistantship (must also register for 405-TA or 406-RA), or registration as a full-time continuing student (402/502) working on a thesis, dissertation, project, or internship. Students admitted full-time who register for credits totaling less than full-time will not achieve a full-time status.

Half-time: Five (5) course credits; Two (2) course credits plus a teaching or research assistantship.

Part-time: One (1) to Four (4) course credits, or registration as a part-time continuing student (401/501).

**In most situations, international students must maintain full-time status. In certain circumstances, international students may be authorized for a reduced course load, allowing them to drop below full-time: Contact the International Center with questions or concerns about Enrollment Status.*

Cross-Registration and Graduate School Consortium

During the academic year, full-time graduate students may take one course per semester through cross-registration agreements with Boston College, Boston University, and Brandeis University. A full-time graduate student at Tufts University may also enroll for two graduate courses during any semester at the Fletcher School of Law and Diplomacy, the Friedman School of Nutrition Science and Policy, and the Graduate School of Biomedical Sciences. Cross-registration is possible on a space available basis. Students who wish to cross-register should consult with the instructor of the course, and should expect to satisfy any prerequisites typically required for enrollment. Cross-registration is not permitted in any summer school. Courses satisfactorily completed (B– or better) at one of the three consortium schools (Boston College, Boston University, and Brandeis University) automatically appear on the student's Tufts transcript and may be counted toward degree requirements.

Tufts students who wish to cross register at a consortium school should first consult with their academic advisor and/or the department's graduate studies committee representative before completing the online Cross-Registration Petition Form through SIS. The host institution reserves the right to terminate the student's participation at that institution at any time.

Cross-registration is also offered through the Consortium for Graduate Studies in Gender, Culture, Women, and Sexuality at MIT. This consortium relationship is limited to the specific area of Women, Gender, and Sexuality Studies. For information about course offerings and application materials, visit the GCWS website.

Audits

Graduate students may arrange with an instructor to sit in on a course, but this course will not appear on the academic record.

Dropping a Course

A course for which a student has registered remains on the record unless it has been dropped within the first five weeks of the term. Courses that are dropped after the fifth week but prior to the last day of classes will carry the grade of W and remain on the transcript. Students who are billed per credit should review the refund policy on the bursar's website for details: https://students.tufts.edu/financial-services/billing/tuition-refund-policy/.

Grades

Grades of scholarship are expressed by one of the following letters:

A Superior work

B Meritorious work

C Not acceptable for graduate credit

D Not acceptable for graduate credit

F Failure

P Not acceptable for graduate credit

S, U Grades of S (Satisfactory) and U (Unsatisfactory) may be given by the instructor in special topics courses, courses in supervised teaching, research courses, certain graduate colloquia, certain professional development courses such as Graduate Institute for Teaching (GIFT), Graduate Research Excellence at Tufts (GREAT), thesis courses, and dissertation courses.

The following symbols are also used:

I Incomplete: an indication that more time will be allowed to complete the work, specifically within six weeks of the first day of classes in the subsequent semester (fall or spring only; summer terms excluded).

PI Permanent Incomplete: Students who received an Incomplete and do not complete the work within the stated time will receive a Permanent Incomplete (PI).

W Withdrawn: an indication that a student has been permitted to withdraw from a course after the fifth week of a semester, but no later than the last day of classes.

Y Work not scheduled for evaluation during the current term.

Changes in Course Grades: Statute of Limitations

Effective education requires timely and objective evaluation of students' academic work, using clear, standard, fair, and public criteria. Such standards should be listed in the course syllabus. While criteria differ across disciplines and faculty, and while the ultimate responsibility for setting standards and evaluating performance rests with departments and individual faculty, submitted grades are final and not subject to negotiation. Exceptions are limited to correcting clerical and calculation errors, and correcting deviations from stated criteria. Students have the right to know the basis of a grade, and faculty should be open to that post-semester conversation. Following such conversation, students who believe that an error or deviation remains may appeal to the department chair and, if necessary, subsequently to the graduate dean.

Health Service Requirements

Prior to initial registration, all graduate students must complete an online medical history and provide proof of required immunizations before July 1. Those with missing or incomplete health reports will have an immunization hold placed on their account and will not be allowed to register for classes until the requirements are completed. For more information about the requirements, you can visit our Resources for New Students website. Please send your questions to Immunization Reviewer via the Patient Portal.

Graduate Co-op Program

Tufts University's School of Engineering (SOE) offers a Cooperative (Co-op) Education Program for full time MS students. The objective of this program is to offer Tufts SOE MS students opportunities to pursue real-world work experience, form professional relationships, and to enhance their resumes. The program's main goal is to introduce students to professional experiences that will assist them in their career path. This program helps students get a feel for a company's culture and work environment, and learn to interact with other colleagues in their field.

Benefits of a Co-Op Program

- Provide students with the skills and frameworks to clarify short- and long-term personal, educational, and career goals and to consistently make prudent, informed decisions throughout one's career.
- Help students explore and experience professional employment in areas related to the student's academic program as well as professional interests and personal aspirations.
- Guide students through the development of a modern personal brand including brand statements, marketing documents curation, and effective network cultivation and strategic management.
- Understanding, practicing, and employing best practices around key career competencies in the modern world of work including, but not limited to: interviews, negotiations, workplace professionalism and etiquette, communication with leaders, managers, and colleagues.
- Receive direct and dedicated support from the Assistant Director, Graduate Co-op Advisor and the Tufts Career Center for the entirety of your co-op preparation and employment experience.

How Co-Op Works

Tufts Graduate School of Engineering's co-op program offers students a unique opportunity to integrate their academic knowledge with industry and subject matter expertise to help employers address and solve practical, real-world problems. Participating students may gain up to eight months of full-time paid work experience that is valuable for building and showcasing distinct, technical, professional, and social competencies employers covet and very often offer co-op alumni with a competitive edge for post-graduation employment opportunities.

Eligible MS students may complete one co-op over the course of their degree program. **Students must apply to join the co-op program at the end of their first semester as an enrolled master's student.** After starting their MS program and completing 18 credits, exclusive of seminar credits, that count toward their degree, students may then look for co-op opportunities.

The Graduate Co-op Program follows a cohort model, where admitted students are grouped based on the semester they intend to begin their co-op positions. During the preceding semester, students will engage in intentional, independent, and cohort-shared experiences across all aspects of the Tufts Career Center Career Planning and Management Process, including self-assessment and discovery, strategic opportunity identification and targeting, custom marketing document creation and co-op search, relationship building practice and management, and modern interview and negotiation practice and preparation. Consistent participation in curated professional development offerings, collaborative and supportive cohort working groups, and regularly scheduled meetings with the Graduate Co-op Advisor is expected, and often mandatory. The Career Center and the SOE Graduate School are sincerely invested in our admitted co-op program students' development and execution plan and have built a system of impactful tools, resources, coaching expertise, and insights that are fundamental to not only securing a co-op but also managing all aspects of a career.

In preparation for a co-op search, students must participate in the required career development sessions offered by Tufts Career Center, which provide valuable skills needed to be successful in securing a co-op position that simultaneously aligns with individual needs and preference while also adhering to Tufts standards and ethical guidelines and employer expectation.

As part of the co-op program, eligible MS students are only allowed to accept one position during their time in the program. *There is no guarantee that students who enter the co-op program will secure employment.*

Cost Of Co-Op Program

There is no additional charge for a student to be enrolled in the co-op program.

Graduate Co-op Program Eligibility

Prerequisites that the student must meet to be considered for the program:

1. Must be enrolled in a program that offers the Graduate Co-op Program (see the list of departments below). This program is not available to part time programs, online programs, certificate programs, Ph.D. programs, or Post Baccalaureate programs. Students enrolled in a combined Post-Bacc/MS program or Fifth-Year MS programs are eligible to participate as long as they meet program requirements.
2. Must complete at least 2 full-time semesters as a fully-matriculated and enrolled Master's student. Students must complete a minimum of 9 credits in their first semester to be considered for the program and must complete at least 18 credits towards their MS programs before beginning their co-op. Newly matriculated stu-

dents can only apply for the program and attend the required info session before the end of their first semester as a full-time student.

   a. Post-Bacc/MS students: No bachelor-level classes for at least 2 full-time semesters before beginning their co-op.
   a. Fifth-Year MS students: Must be fully matriculated with a completed BS degree.

3. Must be in good academic standing (See requirements below), enrolled full-time, have had no previous extensions of time or reduced course load accommodations, and have not enrolled in any undergraduate course as an MS student.
4. Must have at least one semester remaining in their MS programs after co-op opportunity .

Academics and the Co-op Program

Students must meet all requirements of the SOE Graduate Handbook and any departmental specific requirements to be eligible and to participate in the program. All students must meet the following criteria:

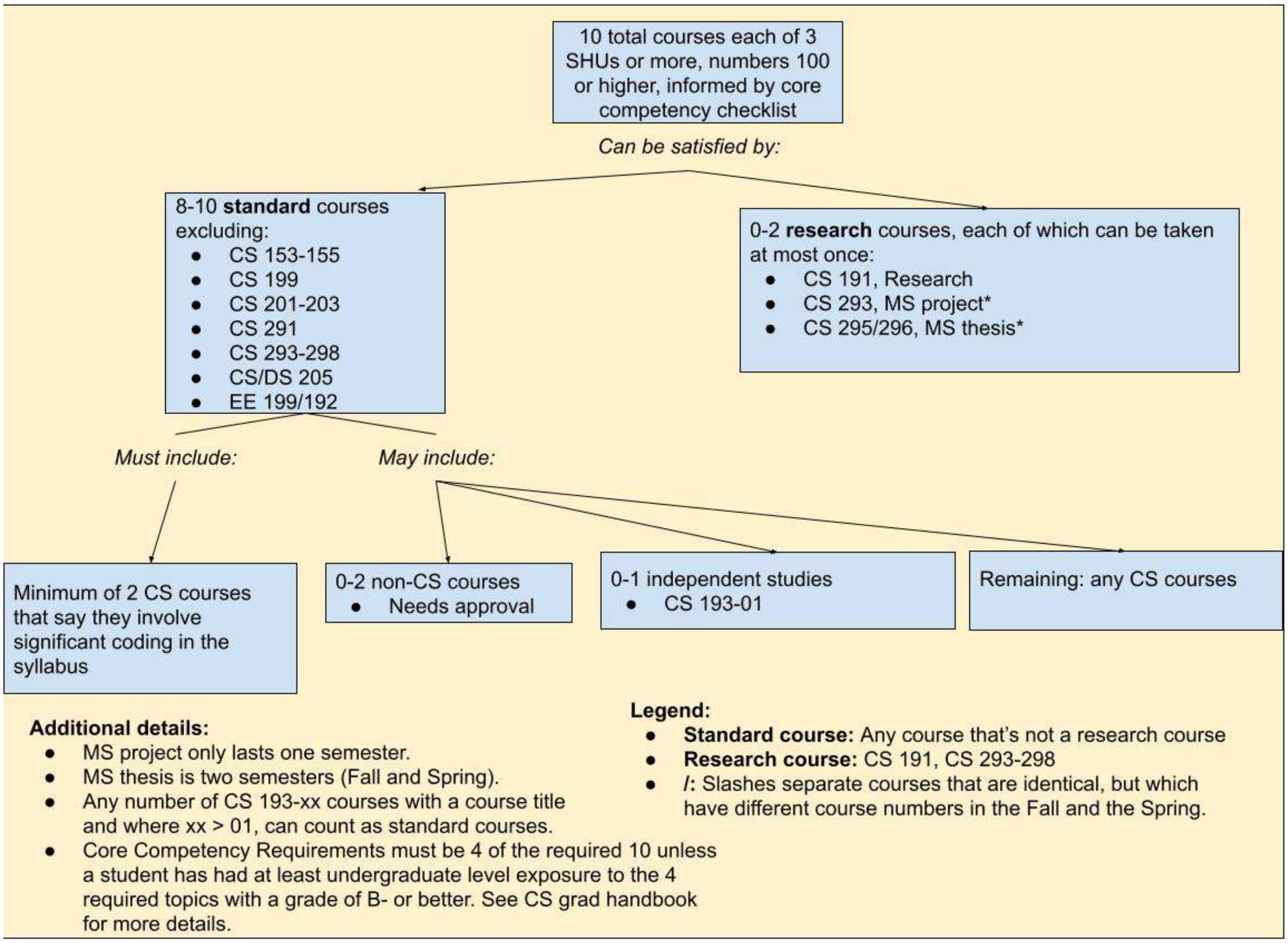
1. Good Academic Standing requirement: Students need to be in good academic standing, enrolled full-time, have had no previous extensions of time or reduced course load accommodations, and have not enrolled in any undergraduate courses as an MS student. Students must be in good academic standing at all times. Failure to remain in good academic standing at all times will result in a student being removed from the co-op program and the MS time to degree will be reduced to the non-co-op time to degree limits for MS programs.
2. Course completion requirement: Fully matriculated students must complete two full-time semesters before working at a co-op, with at least 9 credits total completed in their first semester and at least 18 total credits in the second semester. Courses must be taken at Tufts while matriculated and enrolled as an SOE graduate student. The SOE will be checking to see if an applicant is on track for meeting these requirements in the application process.
3. The student’s academic record must demonstrate a high level of success, including:
   a. No missing grades.
   b. No incomplete grades (grade of I).
   c. No grade of W in the most recent spring/fall semester completed or in progress.
   d. No more than one repeated/substituted course.
   e. No academic probation or disciplinary issues.
   f. No semesters on a reduced course load or degree extensions of time.

Tufts course requirements for a Master's and Ph.D. in Computer Science

This document lists important information about course requirements for Master's and Ph.D. Students. The term "courses" refers to lecture-based classes, independent studies, and research. The first two are denoted by "standard courses" and the latter by "research courses. Both Master's and Ph.D. course requirements can be satisfied via a varying combination of standard and research courses depending on your interests.

We recommend that Master's students interested in completing a thesis and Ph.D. students bias their course selection toward research courses.

M.S. in Computer Science (10 total courses, each of 3 SHUs or more)



10 total courses each of 3 SHUs or more, numbers 100 or higher, informed by core competency checklist

Can be satisfied by:

8-10 **standard** courses excluding:
- CS 153-155
- CS 199
- CS 201-203
- CS 291
- CS 293-298
- CS/DS 205
- EE 199/192

0-2 **research** courses, each of which can be taken at most once:
- CS 191, Research
- CS 293, MS project*
- CS 295/296, MS thesis*

Must include:

Minimum of 2 CS courses that say they involve significant coding in the syllabus

May include:

0-2 non-CS courses
- Needs approval

0-1 independent studies
- CS 193-01

Remaining: any CS courses

Additional details:
- MS project only lasts one semester.
- MS thesis is two semesters (Fall and Spring).
- Any number of CS 193-xx courses with a course title and where xx > 01, can count as standard courses.
- Core Competency Requirements must be 4 of the required 10 unless a student has had at least undergraduate level exposure to the 4 required topics with a grade of B- or better. See CS grad handbook for more details.

Legend:
- **Standard course:** Any course that's not a research course
- **Research course:** CS 191, CS 293-298
- **/:** Slashes separate courses that are identical, but which have different course numbers in the Fall and the Spring.

The flow chart above illustrates the course requirements to get a master's degree. A box that indicates a range in required courses (e.g., 8-10 standard courses) indicates that some of the required courses can be obtained from a box in a sibling branch (e.g., 0-2 research courses).

Core Competencies: By your last semester at Tufts, you must have completed at least one class in each of the four areas listed in Appendix E of the handbook and reproduced below. Designated faculty will hold core competency certification sessions during the first seven days of each semester and can approve and/or advise you on the completion of this requirement.

The competencies can be filled by equivalent classes you may have taken at other universities, and that appear on that university's transcript. Alternatively, you can fill them at Tufts by the courses listed in the sub-bullets below. You must have earned at least a B- in a course, whether at Tufts or elsewhere, to satisfy the relevant course-competency requirement. You will not receive graduate course credit for any course numbered less than 100.

Core Competency areas include:

- Computer Architecture and Assembly Language (CA&AL)
    - CS 40, Machine Structure. *No graduate credit.*
    - CS 111, Operating Systems
    - CS 112, Networks
    - CS 114, Network Security
    - CS 116, Introduction to Security
    - CS 118, Cloud Computing
    - CS 146, (also EE 126) Computer Engineering
    - CS 107 (Formerly COMP/CS 181), Compilers; offered infrequently
    - CS 140, Advanced Topics in Computer Architecture
- Programming Languages (PL)
    - CS 105, Programming Languages
    - CS 21, Concurrent Programming. *No graduate credit.*
    - CS 86, Object-Oriented Programming for GUIs. *No graduate credit.*
    - CS 121 (Formerly COMP/CS 180), Software Engineering
    - CS 107 (Formerly COMP/CS 181), Compilers; offered infrequently
- Data structures and Analysis of Algorithms (DS&AA)
    - CS 160: Intro to Algorithms (we highly recommend taking this class!)
- Theory of Computation (ToC)
    - CS 170, Computation theory
    - If you have little math background, try to take Discrete Math (COMP/CS 61) first.

CS 191: This course is a vehicle for doing research. It has similar requirements to the M.S. project (see below). This course can be taken at most once.

CS 199 (Internship in Computer Science): This course is a vehicle for international students to complete an internship. It does not count towards the 10 course requirement. Reach out to Professor Ming Chow for more information on this course.

M.S. Thesis: The thesis requires a commitment of two semesters total, recorded by enrolling in CS 295 and CS 296 in either order; the M.S. thesis is completely optional. Acceptance to the thesis track occurs after matriculation into the program and only with the support of a faculty advisor who is interested in supervising thesis work. After finding a faculty member who is willing to work with you on a MS Thesis, the faculty member can write to the CS Graduate coordinator to request your change to the thesis track. Some reasons for deciding to do a M.S. thesis may include: 1) you are a M.S. student who wants significant research experience; 2) You want to leave the Ph.D. program with a master's and retain some official record of your research activity; 3) You want to complete a substantial and polished preliminary research project on the way to a Ph.D. You need to submit a thesis prospectus at the end of the first semester and the thesis document just after the end of classes during the semester in which you defend. The deadline can be confirmed here. The defense should be scheduled two weeks earlier than the deadline.

The student and advisor will jointly select a thesis committee subject to approval by the CS Graduate Committee. This committee must include at least three faculty members, including one member from outside the department. You will defend the research via a 45-minute presentation, which the committee will attend.

The defense is also open to the public. You will receive a grade for CS 295 and CS 296 only after you finish both semesters.

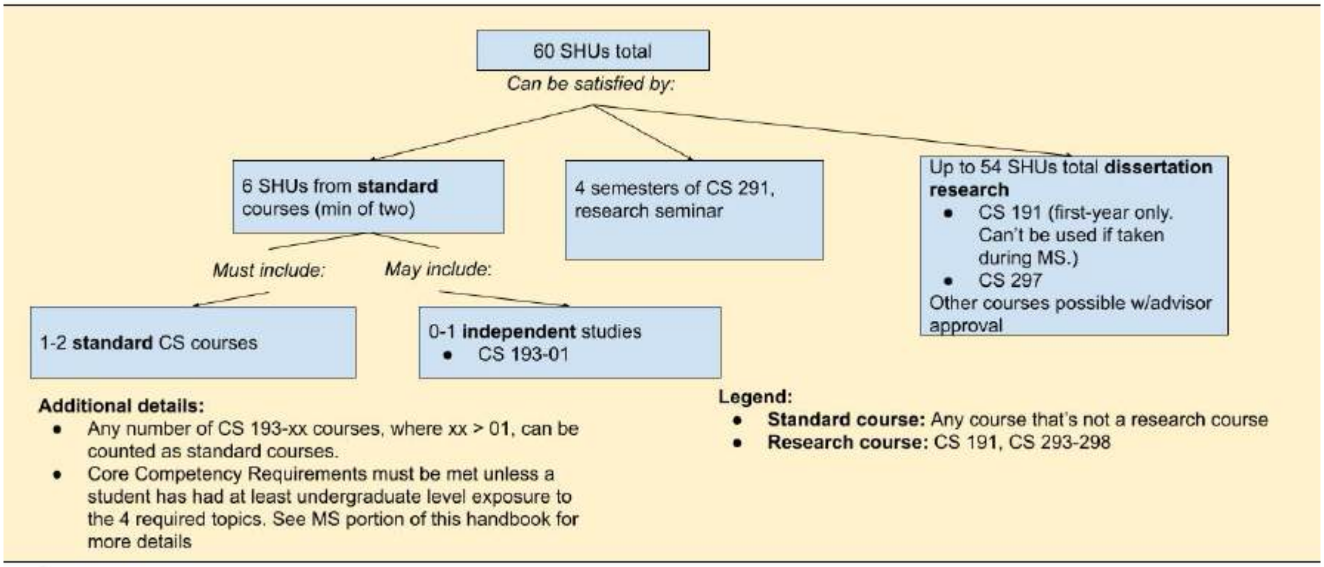
M.S. Project: An M.S. Project consists of research conducted with a faculty advisor and is a commitment of one semester recorded by registering for CS 293, usually for 3 SHUs. You may choose to take this option because: 1) you are a M.S. student who wants to complete a research project without the overhead of writing a thesis, or 2) you are a Ph.D. student who doesn't want to write a separate master's thesis. If you do not have an advisor for the project, you need to find one. If your advisor for the project is not your academic advisor, they need to agree to become your advisor. You and your advisor need to agree on the project and any write-up requirements. (Advisors approve the write-up and keep a copy). This course can be taken at most once.

CS 193-xx (where xx > 01): These are directed study courses with official names and with two or more students. Your advisor might create one as a vehicle for reading a set of research papers from a conference or understanding a new research area. Another faculty member might create one that involves a joint implementation project, or as a dry run of a course that will become a CS 150. You can count any number of these as *standard* CS courses.

Maintaining Good Standing: You must maintain a grade average of at least a B, earning no more than one grade below B-, and make continuous progress toward graduation. Courses that do not count toward your degree requirements must still meet the B- grade requirement.

Applying for Graduation: Graduation information for graduate students can be found at this website. The chart at the bottom of the page outlines what needs to be done by when for each possible graduation date. https://students.tufts.edu/registrar/make-request/apply-graduation/graduation-information-graduate-students.

Ph.D. Requirements


60 SHUs total
Can be satisfied by:
6 SHUs from **standard** courses (min of two)
4 semesters of CS 291, research seminar
Up to 54 SHUs total **dissertation research**
• CS 191 (first-year only. Can't be used if taken during MS.)
• CS 297
Other courses possible w/advisor approval
Must include:
May include:
1-2 **standard** CS courses
0-1 **independent** studies
• CS 193-01

Additional details:

- Any number of CS 193-xx courses, where xx > 01, can be counted as standard courses.
- Core Competency Requirements must be met unless a student has had at least undergraduate level exposure to the 4 required topics. See MS portion of this handbook for more details

Legend:

- **Standard course:** Any course that's not a research course
- **Research course:** CS 191, CS 293-298

The flow chart above shows the course requirements for obtaining a Ph.D. If you don't already have an M.S. degree in computer science or an approved alternative, you must complete these requirements in addition to those for the master's.

Teaching Assistantship: You must TA at least one course during your time as a student at Tufts.

Core Competencies: By the time you take quals (see below), you must certify that you do have background in the areas listed in Appendix E of the handbook. (See comments on core competencies in the M.S. section above.)

- Students without their M.S. must have core competencies finished by the end of their third or fourth semester, and before taking the qualifying exam.
- Students with their M.S. must have core competencies finished by the end of their first or second semester, and before taking the qualifying exam.

Qualifying Exam: This is a sanity check to ensure you are making research progress and have adequate background about your research area. The exam involves giving a presentation about some research you've conducted + an oral exam on 4-7 research papers. The presentation is 30-40 minutes followed by questions. The oral exam is one hour long. You may read more about the process here.

- *Timing:*
  - Students without M.S.: Take it during your third or fourth semester from entry into the program, after satisfying your core competencies. You must pass it by the end of your fifth semester at Tufts.
  - Students with M.S.: Take it by the end of your second semester, after satisfying your core competencies. You must pass it by the end of your third semester at Tufts.
- *Process:*
  - Students, in conjunction with their advisor, select a committee of at least three members. At least two must be insiders of the student's research area, and at least one of these must be a regular faculty member in the computer science department. In addition, there must be at least one member from outside of the student's research area who is a tenured regular member of the computer science faculty. The Grad Committee approves quals committees; they make the final determination of what committees are acceptable.

    - Insider committee members choose 4-7 papers related to the student's research and informs the student of them. These are the papers the student will be evaluated on during the oral exam.
    - Students work with CS Grad Coordinator to schedule both the presentation and the oral exam with the committee. These may be done back-to-back on the same day or on separate days, so long as the research talk occurs first.

Prospectus: You must write a document describing the research you plan to conduct for your dissertation and submit it to the CS graduate committee. The prospectus you submit should be about 2-3 pages long and it must: (1) have a title, (2) describe your intended research direction or open problems to be addressed in the thesis research, (3) cite and briefly describe appropriate related work, (4) identify the dissertation advisor, and (5) identify two additional dissertation committee members within the CS department. Two more members will be added later, (6) include references on any cited work.

- *Process:*
    - Write the prospectus with input from your Ph.D. advisor.
    - Ask two additional Tufts faculty members apart from your advisor who will serve on your committee. List them in the prospectus.
    - Submit the prospectus to the graduate committee six months after your quals. The document should be **signed by your advisor prior to submission**.
    - Your prospectus is a living document and should be updated at least once per year at the time of the grad reviews.

Dissertation Committee: One year after the submission of the prospectus, the student will convene a meeting of the 3 Tufts CS members and the 1 Tufts member outside of CS to review the progress and plans. Six months before the defense, the full committee, including the member external to Tufts, shall meet to map out the expectations for the dissertation.

Dissertation Defense: This is when you are done. During a dissertation defense, you give a public presentation on your research, and then answer private questions from your committee members about both the presentation and the dissertation document that describes your research. The final deadline for submission of the approved dissertation document is just after the end of classes in each of the Spring, Summer, and Fall semesters, and can be confirmed here. The defense occurs two weeks earlier than the university deadline to allow for edits requested by the committee at the defense.

- *Process:*
    - Together with your advisor, propose a committee to the Grad Coordinator. This goes for review to the CS Grad Committee.
    - Convene the committee one year after submitting prospectus and again 6-12 months before defense.
    - Write the dissertation document.
    - Schedule a defense date with your committee.
    - Submit the abstract and title for the dissertation to the CS office at least three weeks before the defense date so that the public portion of the defense can be publicized.
    - Submit the full draft of your dissertation to your committee at least three weeks before the defense date so that they have adequate time to review and to provide you with comments.
        - At the same time, submit a copy to the Graduate Program Coordinator for your student file. It will be made available to faculty or students upon request.
    - Give your defense!
    - Submit final approved document to the university.
- *Committee:*
    - Your committee should have five members.

        - CS Faculty Advisor (with or without tenure)
        - CS Faculty Member (with tenure)
        - CS Faculty Member (with or without tenure)
        - Tufts Faculty Member Outside of CS (does not need to be tenured, can have a joint appointment in CS so long as primary appointment is elsewhere)
        - Member Outside of Tufts (doctoral-level researcher whether in university or industry)
            - This member does not need to be tenured

Applying for Graduation: Graduation information for graduate students can be found at this website. The chart at the bottom of the page outlines what needs to be done by when for each possible graduation date. https://students.tufts.edu/registrar/make-request/apply-graduation/graduation-information-graduate-students.

FAQs for Ph.D. Registrations

Registration for students on RAships and TAships:

- You should register for at least 9 units of load to be considered full-time. You should not register for more than 13 units of load.
- You should register for CS 405-TA or CS 406-RA to indicate status as a Ph.D. TA or RA. These are special courses that count as 3 units of load, but do not count as credit.
- You need to register for at least 6 more SHUs of courses (standard or research) to be considered full time. You should not register for more than 10 additional SHUs of courses.
- An example full-time course load for a Ph.D. student on RA or TAship could be:
  - CS 405-TA/406-RA (3 units of load)
  - CS 135 Machine Learning (3 SHUs)
  - CS 297 Dissertation Research (3 SHUs)
  - Total: 9 units of load, 6 SHUs towards degree requirements

What research courses do I register for once I have completed the 60 SHUs required for the Ph.D.?

- Once you have accrued 60 SHUs, you switch to "CS 502: Matriculation Continued" rather than registering for more research SHUs.
- An example full-time course load for a Ph.D. student who has met the 60 SHU requirement could be:
  - CS 405-TA (3 units of load)
  - CS 502 Matriculation Continued
  - Total: full-time status met

What is the minimum number of "actual" classes I need to take to get an M.S. + Ph.D.?

- You will need to take 8 "actual" classes. Your M.S. would consist of 7 actual courses, 2 research courses (i.e., CS 191 and 293 or CS 295 and 296), and 1 independent study (CS 193-01).
- Your Ph.D. would consist of 1 actual course, 1 CS 193-01, and the rest would be research credits (191, 297). It is possible this number could be further reduced by taking named CS 193-0x classes (x>1), as these count as "actual" classes.

What should I register for if I'm here over the summer?

- Current students who stay for the summer should register for CS 406-RA/405-TA as well as CS 502 – Doctoral Degree Continued.
- A Ph.D. student who has not yet completed the M.S. degree should register for CS 406-RA/405-TA and CS 401/402 – Master's Degree Continued.
- For incoming Ph.D. students who will be here on a temporary visa, they must have a full-time enrollment of 6 SHUs over the summer. This could include one "standard" course, one "research/independent study" course, plus CS 406-RA/405-TA.

32 Results

Arts, Sciences, and Engineering, Summer 2025, CS

□ Show Descriptions □ Show Sections **Enrollment Status:** ● open ⊘ closed ⚠ waitlist

CS-0011 **Introduction To Computer Science**

The study of computer science centers on two complementary aspects of the discipline. First, computer science is fundamentally concerned with the problem-solving methodologies it derives from its foundational fields: the design principles of engineering, mathematical theory, and scientific empirical study. Second, these methodologies are applied in the complex context of a modern day computing system. In this course we will address both of these important aspects. As a means for developing your design skills, we will discuss the fundamental features of a high level, general purpose programming language — namely C++ — and learn how to use it as a tool for problem solving. We will also consider the performance of solutions, and how to apply both analytical and empirical assessment techniques. Finally, we will explore the Unix operating system as a context for problem solving.

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50421	S12	Time Not Specified Online	STAFF	4	●	
C2-LEC Details	50456	S12	Time Not Specified Online	STAFF	4	●	
M1-LEC Details	50490	S12	Time Not Specified Online	STAFF	4	●	
M2-LEC Details	50476	S12	Time Not Specified Online	STAFF	4	●	

CS-0015 **Data Structures**

A second course in computer science. Data structures and algorithms are studied through major programming projects. Topics include linked lists, trees, graphs, dynamic storage allocation, and recursion. Enrollment priority given to freshmen or sophomores; computer science majors or minors; or majors or minors that list CS15 as a requirement or elective.

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50422	S12	Time Not Specified Online	STAFF	4	●	
C2-LEC Details	50423	S12	Time Not Specified Online	STAFF	4	●	
M1-LEC Details	50477	S12	Time Not Specified Online	STAFF	4	●	
M2-LEC Details	50491	S12	Time Not Specified Online	STAFF	4	●	

CS-0061 **Discrete Mathematics**

(Cross-listed as MATH 61). Sets, relations and functions, logic and methods of proof, combinatorics, graphs and digraphs. Recommendations: MATH 32 or COMP 11 or permission of instructor.

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50424	S12	Time Not Specified Online	STAFF	3	●	
C2-LEC Details	50457	S12	Time Not Specified Online	STAFF	3	●	
C3-LEC Details	50460	S12	Time Not Specified TBA Medford/Somerville	STAFF	3	●	
M1-LEC Details	50478	S12	Time Not Specified Online	STAFF	3	●	

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
M2-LEC Details	50479	S12	Time Not Specified Online	STAFF	3	●	

CS-0099 **Internship Computer Science**

Study of approved topics in Computer Science in concert with an internship in computing or a related field. Prerequisites: Permission of instructor.

INTERNSHIP

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
CPT-INT Details	50405	S12	Time Not Specified Online	Ming Yan Chow	1	●	

CS-0111 **Operating Systems**

(Cross-listed as EE 128). Fundamental issues in operating system design. Concurrent processes: synchronization, sharing, deadlock, scheduling. Relevant hardware properties of uniprocessor and multiprocessor computer systems.

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50428	S12	Time Not Specified Online	STAFF	3	●	
M1-LEC Details	50480	S12	Time Not Specified Online	STAFF	3	●	

CS-0115 **Database Systems**

Fundamental concepts of database systems, including conceptual design, relational and object-oriented data models, query languages (SQL, QBE), and implementation issues (indexing, transaction processing, concurrent control). The concepts and algorithms covered encompass many of those used in commercial and experimental database systems. Other topics include distributed databases and distributed query processing. Recommendations: CS 40

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50427	S12	Time Not Specified Online	STAFF	3	●	
M1-LEC Details	50481	S12	Time Not Specified Online	STAFF	3	●	

CS-0120 **Web Programming and Engineering**

Web applications as complex systems that deliver functionality to a large number of users, and exhibit unique behaviors and demands in terms of performance, scalability, usability, and security. How the web works, limitations of client-side and server-side technologies including frameworks and APIs, content optimization, and data persistence and storage. Projects will involve search, using the cloud infrastructure, location-based services, mobile web development, and using tools to assess the security and privacy of web applications.

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50429	S12	Time Not Specified Online	STAFF	3	●	
M1-LEC Details	50482	S12	Time Not Specified Online	STAFF	3	●	

CS-0121 **Software Engineering**

Core principles and ideas that enable development of large-scale software systems, with a focus on programming. Abstraction, modularity, design patterns, specification, testing, verification, and debugging.

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50430	S12	Time Not Specified Online	STAFF	3	●	
M1-LEC Details	50483	S12	Time Not Specified Online	STAFF	3	●	

CS-0131 **Artificial Intelligence**

History, theory, and computational methods of artificial intelligence. Basic concepts include representation of knowledge and computational methods for reasoning. One or two application areas will be studied, to be selected from expert systems, robotics, computer vision, natural language understanding, and planning.

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50431	S12	Time Not Specified Online	STAFF	3	●	
M1-LEC Details	50484	S12	Time Not Specified Online	STAFF	3	●	

CS-0135 **Introduction To Machine Learning And Data Mining**

An overview of methods whereby computers can learn from data or experience and make decisions accordingly. Topics include supervised learning, unsupervised learning, reinforcement learning, and knowledge extraction from large databases with applications to science, engineering, and medicine. Recommendations: CS 160 is highly recommended.

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50432	S12	Time Not Specified Online	STAFF	3	●	
M1-LEC Details	50485	S12	Time Not Specified Online	STAFF	3	●	

CS-0151 **Special Topics in Data Infrastructure and Systems - Cybersecurity Clinic**

A special topics course in data infrastructures and systems, suitable for fulfilling requirements of the Bachelor of Science in Data Science.

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C-LEC Details	50461	S12	Time Not Specified TBA Medford/Somerville	STAFF	3	●	

CS-0160 **Algorithms**

Introduction to the study of algorithms. Strategies such as divide-and-conquer, greedy methods, and dynamic programming. Graph algorithms, sorting, searching, integer arithmetic, hashing, and NP-complete problems. High demand (see "course notes" for signup procedure).

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50425	S12	Time Not Specified Online	STAFF	4	●	
C2-LEC Details	50418	S12	Time Not Specified Online	STAFF	4	●	
M1-LEC Details	50486	S12	Time Not Specified Online	STAFF	4	●	
M2-LEC Details	50487	S12	Time Not Specified Online	STAFF	4	●	

CS-0170 **Computation Theory**

(Cross-listed as MATH 191). Models of computation: Turing machines, pushdown automata, and finite automata. Grammars and formal languages, including context-free languages and regular sets. Important problems, including the halting problem and language equivalence theorems.

LECTURE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-LEC Details	50426	S12	Time Not Specified Online	STAFF	3	●	
M1-LEC Details	50488	S12	Time Not Specified Online	STAFF	3	●	

CS-0191 **Research**

Research on a topic in Computer Science or a related discipline, culminating in a final paper describing accomplishments, with the goal of advancing the state of the art. Topic is proposed by a faculty sponsor in Computer Science. Faculty consent required. Students sign up for a section that corresponds to a faculty member.

RESEARCH

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C-RSC Details	50412	S12	Time Not Specified TBA Medford/Somerville	Abani Patra, Alva Couch, Bert Huang, Daniel Jared Votipka, Dave Lillethun, Deborah Sunter, Diane L Souvaine, Donna Slonim, Elaine Schaertl Short, Ethan E. Danahy, Fahad Rafique Dogar, Jeffrey Foster, Jivko Sinapov, Johannes Peter Albert De Ruiter, Karen A Panetta, Karen Edwards, Lenore J Cowen, Liping Liu, Mark A Sheldon, Marty Allen, Matthias Scheutz, Megan Monroe, Michael C. Hughes, Ming Yan Chow, Noah Mendelsohn, Norman Ramsey, Peter John Love, Raja Raman Sambasivan, Remco K Chang, Richard Townsend, Robert Jacob, Samuel Guyer, Soha Hassoun, Susan Landau	3	●	

CS-0193 **Directed Study**

Guided study of an approved topic. Please see departmental website for specific details.

INDEPENDENT STUDY

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
A-IND Details	50440	SA	Time Not Specified TBA Medford/Somerville	Diane L Souvaine, Elaine Schaertl Short, Jivko Sinapov	0-4	●	

CS-0193 **Directed Study**

Guided study of an approved topic. Please see departmental website for specific details.

INDEPENDENT STUDY

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
B-IND Details	50473	SB	Time Not Specified TBA Medford/Somerville	Diane L Souvaine, Elaine Schaertl Short, Jivko Sinapov	0-4	●	

CS-0193 **Directed Study**

Guided study of an approved topic. Please see departmental website for specific details.

INDEPENDENT STUDY

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C-IND Details	50401	S12	Time Not Specified TBA Medford/Somerville	Abani Patra, Alva Couch, Bert Huang, Daniel Jared Votipka, Dave Lillethun, Diane L Souvaine, Donna Slonim, Elaine Schaertl Short, Ethan E. Danahy, Fahad Rafique Dogar, Jeffrey Foster, Jivko Sinapov, Johannes Peter Albert De Ruiter, Karen Edwards, Lenore J Cowen, Liping Liu, Mark A Sheldon, Marty Allen, Matthias Scheutz, Megan Monroe, Megumi Ando, Michael Allan Jahn, Michael C. Hughes, Milod Kazerounian, Ming Yan Chow, Noah Mendelsohn, Norman Ramsey, Raja Raman Sambasivan, Remco K Chang, Richard Townsend, Robert Jacob, Samuel Guyer, Soha Hassoun, Susan Landau	0-4	●	

CS-0193 **Directed Study - Algorithms Practicum**

Guided study of an approved topic. Please see departmental website for specific details.

INDEPENDENT STUDY

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C1-IND Details	50470	S12	Time Not Specified TBA Medford/Somerville	Michael Allan Jahn	0-4	●	

CS-0193 **Directed Study - Discrete Mathematics**

Guided study of an approved topic. Please see departmental website for specific details.

INDEPENDENT STUDY

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C2-IND Details	50469	S12	Tu, Th 1:30PM - 3:00PM No Room Assigned Medford/Somerville	Michael Allan Jahn	0-4	●	

CS-0288 **Master of Science Capstone Project I**

Part one of a two-course, hands-on, and project-based culmination to the Master of Science in Computer Science Online program. Application of principles, strategies, methods, and tools for requirements analysis and design of a programming project, including project planning, project management, and proof of concept prototyping. Formulation of a project plan, including estimation of project completion requirements and timeline. To be taken in the second-to-last term of the Master of Science in Computer Science Online degree. Not available to students outside that program. Prerequisites: CS 180 or 121, and enrollment in the Master of Science in Computer Science Online program.

INDEPENDENT STUDY

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
M1-IND Details	50475	S12	Time Not Specified Online	Marty Allen	3	●	

CS-0289 **Master of Science Capstone Project II**

Part two of a two-course, hands-on, project-based culmination experience for the Master of Science in Computer Science Online program. Implementation of the project defined in part one, including use of principles, tools, and strategies for implementation, debugging, testing, documentation, maintenance, and release management. Presentation of final project results and documentation. To be taken in the last term of the Master of Science in Computer Science Online degree. Not available to students outside that program. Prerequisites: CS 288, and enrollment in the Master of Science in Computer Science Online program.

INDEPENDENT STUDY

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
M1-IND Details	50416	S12	Time Not Specified Online	Marty Allen	3	●	

CS-0293 **Master's Project**

Guided individual study of an approved topic suitable for a master's design project. Please see departmental website for specific details. Faculty consent required. Students sign up for a section that corresponds to a faculty member.

PROJECT

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C-PRO Details	50402	S12	Time Not Specified Online	Abani Patra, Alva Couch, Bert Huang, Daniel Jared Votipka, Dave Lillethun, Diane L Souvaine, Elaine Schaertl Short, Fahad Rafique Dogar, Jeffrey Foster, Jivko Sinapov, Johannes Peter Albert De Ruiter, Lenore J Cowen, Liping Liu, Mark A Sheldon, Marty Allen, Matthias Scheutz, Megan Monroe, Michael C. Hughes, Ming Yan Chow, Noah Mendelsohn, Norman Ramsey, Raja Raman Sambasivan, Remco K Chang, Richard Townsend, Robert Jacob, Samuel Guyer, Soha Hassoun, Susan Landau	0-4	●	

CS-0295 **Masters Thesis**

Guided individual study of an approved topic suitable for a master's design project. Please see departmental website for specific details. Faculty consent required. Students sign up for a section that corresponds to a faculty member.

THESIS

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C-THS Details	50409	S12	Time Not Specified Online	Abani Patra, Alva Couch, Bert Huang, Daniel Jared	0-6	●	

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
				Votipka, Dave Lillethun, Deborah Sunter, Diane L Souvaine, Donna Slonim, Elaine Schaertl Short, Fahad Rafique Dogar, Jeffrey Foster, Jivko Sinapov, Johannes Peter Albert De Ruiter, Lenore J Cowen, Liping Liu, Mark A Sheldon, Marty Allen, Matthias Scheutz, Megan Monroe, Michael C. Hughes, Milod Kazerounian, Ming Yan Chow, Noah Mendelsohn, Norman Ramsey, Peter John Love, Raja Raman Sambasivan, Remco K Chang, Richard Townsend, Robert Jacob, Samuel Guyer, Soha Hassoun, Susan Landau			

CS-0297 **Dissertation Research**

Guided research on a topic suitable for a doctoral dissertation. Please see departmental website for specific details. Prerequisites: Ph.D. student standing in Computer Science

INDEPENDENT STUDY

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
C-IND Details	50411	S12	Time Not Specified TBA Medford/Somerville	Abani Patra, Alva Couch, Bert Huang, Daniel Jared Votipka, Dave Lillethun, Diane L Souvaine, Donna Slonim, Elaine Schaertl Short, Fahad Rafique Dogar, Jeffrey Foster, Jivko Sinapov, Johannes Peter Albert De Ruiter, Johes Bater, Karen A Panetta, Kathleen Fisher, Lenore J Cowen, Liping Liu, Mark A Sheldon, Marty Allen, Matthias Scheutz, Megan Monroe, Michael C. Hughes, Ming Yan Chow, Noah Mendelsohn, Peter John Love, Raja Raman Sambasivan, Remco K Chang, Richard Townsend, Samuel Guyer, Shuchin Aeron, Soha Hassoun, Susan Landau	1-9	●	

CS-0299 **Internship In Computer Science**

Study of approved topics in Computer Science in concert with an internship in computing or a related field outside the University. Prerequisites: Permission of instructor

INTERNSHIP

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
CPT-INT Details	50406	S12	Time Not Specified Online	Ming Yan Chow	1	●	

CS-0401 **Masters Degree Continuation**

Part-time.Please see departmental website for specific details.

CONTINUANCE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
PT-CON Details	50465	SA	Time Not Specified TBA Medford/Somerville	STAFF	0	●	

CS-0401 **Masters Degree Continuation**

Part-time.Please see departmental website for specific details.

CONTINUANCE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
PT-CON Details	50462	S12	Time Not Specified TBA Medford/Somerville	STAFF	0	●	

CS-0402 **Masters Degree Continuation**

Full-time.Please see departmental website for specific details.

CONTINUANCE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
FTA-CON Details	50463	SA	Time Not Specified TBA Medford/Somerville	Diane L Souvaine	0	●	

CS-0402 **Masters Degree Continuation**

Full-time.Please see departmental website for specific details.

CONTINUANCE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
FT-CON Details	50407	S12	Time Not Specified Online	Diane L Souvaine	0	●	

CS-0405 **Grad Teaching Assistant**

TEACHING ASSISTANT

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
TA-TAS Details	50403	S12	Time Not Specified Online	Abdullah Bin Faisal, Andrew Scott Winslow, Diane L Souvaine, Elaine Schaertl Short, Fabrizio Santini, Jivko Sinapov, Marty Allen, Michael Allan Jahn, Milod Kazerounian, Owen A. Morrissey, Patrick Feeney, Sandra Schulenburg, Shivam Goel	0	●	

CS-0406 **Grad Research Assistant**

RESEARCH

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
RA-RSC Details	50404	S12	Time Not Specified Online	Daniel Jared Votipka, Elaine Schaertl Short, Fahad Rafique Dogar, Jeffrey Foster, Jivko Sinapov, Lenore J Cowen, Matthias Scheutz, Megumi Ando, Michael C. Hughes, Peter John Love, Raja Raman Sambasivan, Remco K Chang, Robert Jacob, Saeed Mehraban, Sandra Schulenburg, Soha Hassoun, Vasanth Sarathy	0	●	

CS-0502 **Doctoral Degree Continuation**

Full-time.Please see departmental website for specific details.

CONTINUANCE

Section	Class No.	Session	Day, Times and Locations	Faculty	Credit	Status	Select
CON-CON Details	50408	S12	Time Not Specified Online	Diane L Souvaine	0	●	

    g. Satisfied language assessment and completion of the English for Technical Professionals online course (for international students who were required to submit TOEFL/language scores in their application).

4. Students searching for a co-op position during a fall or spring semester must meet the semester-hour requirements.
5. All students must have their academic advisors sign off on the Co-op Agreement form to ensure a return plan that outlines a feasible pathway for completing the remaining requirements within the degree time limits. The sequence of courses in the program should be considered in addition to possible alternatives to required courses, should a required course not be offered during the semester when the student returns from co-op. A student's degree program cannot be extended because a required course is not offered during the semester that they return.
6. After the co-op, students must return to complete one full semester at Tufts before graduating. There must be degree requirements remaining in their program after the co-op is completed. Students cannot go on a co-op if all degree requirements are already satisfied.

Departments Participating in Co-Op Program

Students enrolled in MS degree programs offered by the following departments are eligible to participate in the co-op program for graduate students. Part-time, online certificate, post-bacc, and Ph.D. students students do not qualify for the co-op.

- Biomedical Engineering
- Chemical and Biological Engineering
- Civil and Environmental Engineering
- Computer Science
- Electrical and Computer Engineering
- Mechanical Engineering
- Tufts Gordon Institute

Questions and Contact Information

Graduate Admissions for new applicants.

Career Center for enrolled MS students.

Co-op Application Process

To be considered for enrollment in the Graduate Co-op Program, students must:

1. **Attend a REQUIRED Graduate Co-op Information Session.** Info sessions are typically held at the end of the semester and are hosted by the Career Center and partnership with SOE. Eligible students will receive an announcement to their

CS-119 Syllabus for Spring, 2025

Teaching Team

Professor	Jitendra Singh
Assistants	TBA

On-Campus Lectures: Mondays and Wednesdays 10:30 am — 11:45 am. Joyce Cummings Center, Room 180, except where noted below[1].
Online Synchronous Meetings: Mondays 5:30 — 7:00 pm, except where noted below[2].

The meeting dates for both sections are shown below[3], (note the exceptions for week of 2/17)

Wk-Of	1/15	1/22	1/27	2/3	2/10	2/17	2/24	3/3	3/10		3/24	3/31	4/7	4/14	4/23	4/28
Day	W	W	M W	M W	M W	W Th	M W	M W	M W		M W	M W	M W	M W	W	M
Campus	✅	✅	✅✅	✅✅	✅✅	✅✅	✅✅	✅✅	✅✅		✅✅	✅✅	✅✅	✅✅	✅	✅
Online	x	✅	✅ x	✅ x	✅ x	x ✅	✅ x	✅ x	✅ x		✅ x	✅ x	✅ x	✅ x	x	✅

[1] Campus lectures on 1/15 and 1/22 will also be available on zoom (requires Tufts ID to access) and recorded for the benefit of students joining before the last day for AS&E students to ADD classes.
[2] Online lectures will begin on 1/22. All online lectures will be recorded.
[3] Per the University calendar.

CS-119 Syllabus for Spring, 2025

Course Description

Big Data deals with emerging applications in science and engineering disciplines that generate and collect data at unprecedented speed, scale, and complexity — and the techniques for these data to be processed and analyzed efficiently.

CS-119 introduces the latest techniques and infrastructures developed for big data including parallel and distributed database systems, map-reduce infrastructures, scalable platforms for complex data types, stream processing systems, and cloud-based computing.

The course content will be a blend of theory, algorithms and practical (hands on) work, involving software design, coding, testing and debugging!

Prerequisites: (1) Linear Algebra and (2) Fluency with Python. There are no other formal requirements but it helps if you enjoy programming, especially debugging code! Familiarity with database internals (CS-115) is helpful but is not required. Most of our work in this class will be Python-based. We'll primarily (but not exclusively) use Google Colab.

Big Data work across the industry involves Python, Java, Scala, shell programming and SQL. Familiarity with the programming languages cited will give you a head start. It is expected that students taking CS-119 know at least some of these and will pick up the rest, as required, on their own.

About the "Alice" Theme: Some of the exercises in this course playfully evoke *Alice in Wonderland* by Lewis Carroll[4].

[4] In addition to authoring Alice in Wonderland, Lewis Carroll was a Mathematician. He is credited with a paper on infinite logic, What the Tortoise said to Achilles, popularized by Douglas Hofstadter's writings.

Hofstadter's *I am a Strange Loop* and its predecessor *Gödel, Escher, Bach (GEB)* are a mixture of his musings on consciousness, intelligence (human and artificial), mathematics and a whole lot more. GEB won the Pulitzer Prize for General Non-Fiction in 1980. The Strange Loop Conferences, sadly, ended in 2023.

CS-119 Syllabus for Spring, 2025

Textbooks on Big Data Theory

The Datacenter as a Computer *Designing Warehouse-Scale Machines*, Third Edition Luiz André Barroso Urs Hölzle Parthasarathy Ranganathan	Springer Cham Copyright: 2019 ISBN: 978-3031006333 Downloadable from Tufts Library with your credentials.
Mining of Massive Datasets, 3rd edition Jure Leskovec Anand Rajaraman Jeff Ullman	Cambridge University Press Copyright: 2014 ISBN: 978-1108476348 Textbook home page
Everybody lies *Big data, new data, and what the internet can tell us about who we really are* Seth Stephens-Davidowitz	Harper Collins Copyright: 2017 ISBN: 9780062390875

Textbooks on Big Data Programming

Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale, 4th Edition Tom White	O'Reilly Media Copyright: 2015 ISBN: 978-1491901632
Spark: The Definitive Guide Bill Chambers Matei Zaharia	O'Reilly Media Copyright: 2018 ISBN: 978-1491912218

Grades for the Course[5]

Item	% score
Class Participation Your class participation score is intended to reflect your effort in maintaining a collaborative learning environment for everyone. ● Thoughtful public postings on Piazza, our class discussion platform, ● Willingness to help peers when they are stuck and being acknowledged for having done so, ● Participation during office hours & lectures and – for the online section – turning on your video camera and keeping it turned on!	**8%**
Exercises The exercises are intended to reflect your effort in staying current with what's happening in class. They are typically due a week after they are assigned, never more than two weeks. Only the best 7 out of 9 of your exercise scores will count towards the final grade. However, not all exercises are equal – two of the 9 exercises are worth more points than the other 7.	**55%**
Assignments Big Data is a continuously evolving field. The ability to consume published, peer-reviewed papers is critical to your ability to stay current with it. We will follow a methodology proposed by S. Keshav to review some of the seminal papers in the field. Small programming assignments may be utilized to prepare for forthcoming exercises.	**17%**
Projekt[6] Writing proposals is an integral part of moving up in the organization where you work, be it academe or industry. The first step for a research project is to apply for funding. The projekt is essentially that: a funding proposal. You will not be doing the actual research just yet — you will be creating a proposal[7] for securing funding for the research. Choosing a topic thoughtfully is part of the project.	**20%**

[5] All Exercises, Assignments and the Projekt are open-book, open-internet, take home. There is no midterm or final exam.

[6] Short for "GedankenProjekt." It is named in honor of "Gedankenexperiment," a term used by Albert Einstein to describe his unique approach of using conceptual rather than actual experiments in creating the theory of relativity.

[7] A research proposal is a detailed plan or 'blueprint' for the intended study, and once it is completed, the research project should flow smoothly. The Projekt will include the technical design of a system but not its implementation.

Course Administration and Policies

Lab Environment

Each student will have an allowance for use of a Cloud Platform to be used for some of the earlier exercises. Google has been generous in making this resource available to us through their "Google Cloud Platform credits" program (GCP credits program), please be thoughtful in using it. Some things to consider:

- *Shut off your VM or cluster when you are done using it, otherwise they will continue to accrue charges.*
- *Don't confuse Google's "Free Trial" program with the GCP credits program* and be aware of the risk of signing up for the Free Trial. It is deceptively easy to change to billing that credit card when the $300/90 days of the Free Trial is used up.
- *If your credit card is charged at the end of the Free Trial program, GCP credits admins cannot help.* Please see this warning from the GCP credits program.

Seeking Help

Please keep in mind the following as regards to approaching the instructor for help.

- For technical questions, please utilize Piazza — plus other students could likely benefit from the Q & A. Some students may answer your questions even faster! *Public questions will receive a higher weighting because they apply to everyone.*
- For questions that have specifics of your solution that you don't want others to see, it's OK to post private questions.
- To get in touch with the instructor for a matter unrelated to course content, please use email. Please keep the use of email to confidential matters, not for general class discussion
- If there is no response from the above within 24 hrs, or in case of an emergency, please call the instructor.

Illness-related policies

Please do not come to class when exhibiting even mild Covid-19 symptoms. If you are ill or symptomatic, please alert the teaching team via Piazza. This guideline applies to everyone: the students, TAs and the instructor.

Zoom recordings will be available to those who are unable to attend due to illness, anxiety, grief or trauma. On-campus students must inform the instructor via email at least 2 hours prior to class.

To make zoom recordings available to everyone for unexpected absences, please be aware that you are consenting to being recorded, even if we don’t record on a particular day..

Academic Integrity

You are expected to be familiar with the Student Guide to Academic Integrity at Tufts and follow those guidelines.

Accommodations for Students with Disabilities

Tufts University values the diversity of our students, staff, and faculty and recognizes the important contribution each student makes to our unique community. Tufts is committed to providing equal access and support to all qualified students through the provision of reasonable accommodations so that each student may fully participate in the Tufts experience. If you have a disability that requires reasonable accommodations, please contact the StAAR Center (formerly Student Accessibility Services) at StaarCenter@tufts.edu or 617-627-4539 to make an appointment with an accessibility representative to determine appropriate accommodations. Please be aware that *accommodations cannot be enacted retroactively*, making timeliness a critical aspect for their provision.

Academic Support at the StAAR Center

The StAAR Center (formerly the Academic Resource Center and Student Accessibility Services) offers a variety of resources to all students (both undergraduate and graduate) in the Schools of Arts and Science, Engineering, the SMFA and Fletcher; services are free to all enrolled students. Students may make an appointment to work on any writing-related project or assignment, attend subject tutoring in a variety of disciplines, or meet with an academic coach to hone fundamental academic skills like time management or overcoming procrastination. Students can make an appointment for any of these services by visiting the StAAR Center website.

Mental Health Support

As a student, there may be times when personal stressors or emotional difficulties interfere with your academic performance or well-being. The Counseling and Mental Health Service (CMHS) provides confidential consultation, brief counseling, and urgent care at no cost for all Tufts undergraduates as well as for graduate students who have paid the student health fee. To make an appointment, call 617-627-3360. Please visit the CMHS website to learn more about their services and resources.

About the Instructor

I received my M.S. in Electrical Engineering with a thesis on handwriting recognition – a topic for which we now use neural networks. My Ph.D. in Electrical Engineering was on solving large-scale matrix problems in Electromagnetics. A significant part of my career was in industry, mostly in Systems Architect roles, Computer-Aided Design at first and later in Finance.

I have been a member of the faculty at WPI in the past, and more recently at Tufts. I've taught Big Data for 10+ years. I have worked in Cloud Computing, Big Data and Python since 2008. Python has been my programming language of choice ever since!

Please call me Jitendra or J or Prof. J, whichever you prefer. (No period after the J)

ARTIFICIAL INTELLIGENCE

Prof. Fabrizio Santini - Fabrizio.Santini@tufts.edu

This course is an introductory survey of Artificial Intelligence (AI). It will cover AI's history, theory, and computational methods. Basic concepts include the representation of knowledge and computational methods for reasoning.

TUFTS: http://www.cs.tufts.edu/comp/131/

CANVAS: https://canvas.tufts.edu/courses/63214

- **Text**: Artificial Intelligence: A Modern Approach (4th edition). Stuart Russell and Peter Norvig, Prentice Hall (2021) ISBN: 0-13-461099-7
- **Hours:** Monday – Wednesday, 6 PM – 7:15 PM, Joyce Cummings Center 160
- **Block:** M+
- **Recorded classes:** Classes are not recorded.
- **Prerequisites:** Algorithms and data structures, basic Linear Algebra, basic Probability Theory.
- **Office hours:** After every class or by appointment.

COURSE GOALS

By the end of the semester, students should be able to:

1. Identify the major classical and modern AI paradigms and explain how they relate to each other.
2. Analyze the structure of a given problem such that they can choose an appropriate paradigm in which to frame that problem.
3. Implement a wide variety of both classical and modern AI algorithms.

TENTATIVE LIST OF TOPICS

The following is a tentative list of topics covered in this course. The list will be subject to changes that depend on the time and pace of the class:

01/15 **Wed**	Introduction to AI
01/20 **Mon**	**NO CLASS**
01/22 **Wed**	Rational agents
01/27 **Mon**	Behavior Trees
01/29 **Wed**	Uninformed search
02/03 **Mon**	Informed search 1
02/05 **Wed**	Informed search 2
02/10 **Mon**	Constraint Satisfaction Problems
02/12 **Wed**	Local search
02/17 **Mon**	**NO CLASS**
02/19 **Wed**	**Test 1** (6:00 PM – 7:15 PM, Joyce Cummings Center 160)
02/20 **Thu**	Propositional Logic 1
02/24 **Mon**	Propositional Logic 2
02/26 **Wed**	First-order Logic 1
03/03 **Mon**	First-order Logic 2
03/05 **Wed**	Probability Theory 1
03/10 **Mon**	Probability Theory 2
03/12 **Wed**	**Test 2** (6:00 PM – 7:15 PM, Joyce Cummings Center 160)
03/17 **Mon**	**NO CLASS**
03/19 **Wed**	**NO CLASS**
03/24 **Mon**	Bayes networks 1
03/26 **Wed**	Bayes networks 2
03/31 **Mon**	Markov models 1
04/02 **Wed**	Markov models 2
04/07 **Mon**	Learning by examples
04/09 **Wed**	Artificial Neural Networks 1
04/14 **Mon**	Artificial Neural Networks 2
04/16 **Wed**	Clustering
04/21 **Mon**	**NO CLASS**
04/23 **Wed**	Cognitive architectures
04/28 **Mon**	Ethics in AI
05/08 **Thu**	**Test 3** (12:00 PM – 2:00 PM, Joyce Cummings Center 160)

HOMEWORK AND TESTS

Assignments

Six (6) assignments (roughly matching the major course sections) will be given during the class.

Python is the official implementation language for assignments. C++ or other languages can be used **ONLY** after negotiating with the teaching staff before submitting.

You can submit a ZIP file with all the files needed to run your solution on Canvas. There is no auto-grader. All assignments are manually executed and graded. For Python, provide a plain PY file (**DO NOT** use `Jupyter` notebooks). If you are writing in C++, please include a `CMakeLists.txt` file and other compilation instructions.
Your solutions may make use of any numerical libraries for pre-processing and visualization. However, the core portion of your solutions **MUST** be implemented from scratch.
Any material regarding the solution will be submitted electronically. Homework is due at midnight of the deadline (check the COMP 131 website calendar for more details). Late assignments are penalized at 10% for each 24-hour delay. No homework will be accepted after one week.

You have three (3) 4-day extensions available to you. Multiple extensions cannot be combined on a single assignment. If you decide to use them, please alert the teaching staff **before the assignment's deadline**.

Homework

For reading assignments, students will be asked to read a section from the textbook. Occasionally, an exercise or two will be given in class to be solved at home. The results of the homework will be discussed upon request from the students.

Attendance

For some lectures, I will provide a short questionnaire that must be returned at the end to promote engagement and obtain feedback on the class's understanding of the topics. The questionnaire is based on best effort, so there is no actual grade. However, it will impact the attendance portion of your grade. You can miss at most two questionnaires without any impact on the grade.

Tests

Three (3) tests will be administered over the semester (see the tentative schedule below). The tests will combine single-choice, multiple-choice, calculations, and open-ended answers. Books must be closed, and electronic devices are not allowed.
If you need to miss any of the tests for any reason, you must inform me **before the scheduled day** so that a make-up session can be arranged.

Please Note: If you wish to dispute a grade, you must do so **within one week** of receiving the grade. After such a term, the grade will be considered final.

Do NOT bring unauthorized materials, information, or any electronic equipment with you to a room where an exam is being administered. **DO NOT** engage in behavior that looks like cheating, such as passing a note to a friend, whispering to another student while the exam is in progress, or looking toward another student's work. **DO NOT** bring your cell phone, tablet, music device, programmable calculator, or any other electronic device to an exam room. If an exam proctor sees you handling an electronic device, even silencing a phone if it rings or vibrates in the middle of the exam, the Judicial Affairs Administrator will treat it as an academic integrity violation. **DO** turn off your cell phone and put it out of reach, out of sight, or as instructed before the exam begins.

Final grades

You must show proficiency in all grading areas to pass the class. A failing average (below 70 or "C-") in *any* of the grading areas (assignments or tests) will result in a failing grade in the class. Your final grade will be determined using the following percentage breakdown: 45% coding assignments, 50% tests, and 5% attendance.

The following standard grading scale will be applied without any grading on the curve:

70 – 72.99	C-
73 – 76.99	C
77 – 79.99	C+
80 – 82.99	B-
83 – 86.99	B
87 – 89.99	B+
90 – 94.99	A-
95 – 98.99	A
99 – 100	A+

COMMUNICATIONS

Canvas will be our primary means of communication. The website will also be the venue for all course announcements and class discussions.

Rather than emailing questions to the teaching staff, we encourage you to post your questions as a public discussion. You are also encouraged to help each other, as long as the question does not contain any code or portion of a problem set answer. In such a case, the question must be made private (please refer to the section Academic Honesty below).

To schedule office hours outside the posted times, please email any teaching staff or post a private email or a message on Canvas.

BEHAVIOR IN CLASS

Please respect your classmates' right to learn without unwanted distractions by silencing your cell phones and other electronic devices before class begins. The teaching staff will be doing the same. **No text messaging**, **instant messaging**, **gaming**, or **web surfing** will occur during class. You may use laptops to take notes during lectures, but a single violation of this policy will result in losing that privilege.
An exception will be made in the case of a disability if the student approaches the instructor beforehand and an arrangement is agreed to.

If you are disruptive in class, the teaching staff will ask you not to be. If you continue to be disruptive, we will ask you to leave.

ACADEMIC HONESTY

Science is, at its core, a collaborative effort. The advantages of coming together for examination or comparison, sometimes even to explain the problem, are well known. I strongly encourage students to discuss course material, problems, and applications outside the classroom with the teaching staff and other students. You are also encouraged to form study groups for the tests.

Integrity and honesty, however, are equally important qualities of any future academic, scientist, or engineer. We take plagiarism very seriously. You must do your homework and the final projects independently and without the help of any AI-enabled tools. If you need help, the teaching staff will be more than happy to help you!

The Faculty of the School of Arts and Sciences and the School of Engineering must report suspected academic integrity violations to the Dean of Student Affairs Office. If I suspect you cheated or plagiarized in this class, I must report the situation to the dean. If students do not understand these terms or those outlined in the Academic Code of Conduct, they must talk to the instructor.

STUDENTS WITH DISABILITIES

To maximize each student's participation in the Tuft experience, Tufts and the teaching staff are committed to providing equal access and support to all qualified students through reasonable accommodations.

If you need reasonable accommodations for a disability, don't hesitate to contact the Student Accessibility Services office at mailto:accessibility@tufts.edu or 617-627-4539 to determine appropriate actions.

Please be aware that accommodations cannot be enacted retroactively, making timeliness a critical aspect of this provision. If you need special accommodations for exams, please do not wait until just before the exam to contact SAS. Please give us sufficient time to arrange the necessary accommodations.

MENTAL HEALTH SUPPORT

As a student, there may be times when personal stressors or emotional difficulties interfere with your academic performance or well-being. The Counseling and Mental Health Service (CMHS) provides confidential consultation, brief counseling, and urgent care at no cost for all Tufts undergraduates and graduate students who have paid the student health fee. To make an appointment, call 617-627-3360. Please visit the CMHS website: http://go.tufts.edu/Counseling to learn more about their services and resources.

FEEDBACK

Tufts and the Artificial Intelligence teaching staff strive to create a learning environment welcoming students of all backgrounds. Your thoughts and concerns are important. You are encouraged to give feedback to the instructor throughout the term. If you feel uncomfortable talking to teaching staff members, consider contacting your academic advisor, department chair, or dean.

COMP 23: Introduction to Game Development

Tufts University Department of Computer Science, Fall 2015

Instructor

- Ming Chow, mchow@cs.tufts.edu
- Office Hours: Wednesdays from 1 - 4 PM, or by appointment, "in my usual spot". Hours are good until the last day of classes, December 11th.
- Please send all class questions (e.g., help on assignments and labs) via Piazza. DO NOT E-MAIL ME! Sign up at https://piazza.com/tufts/fall2015/comp23.
- For emergencies or private matters, please e-mail or see me directly.

Teaching Assistant

- Arthur Berman
- Arthur will hold office hours between 11 and 2 on Fridays. Arthur will be showing when and where he is on office hours on http://www.halliganhelper.com.

Class Time and Location

- Tuesdays and Thursdays, 12:00 - 1:15 PM in Halligan 111A

Prerequisites

- COMP 15: Data Structures

Documentation

- Phaser.io: http://phaser.io/docs
- Unity: http://docs.unity3d.com/Manual/index.html

Syllabus

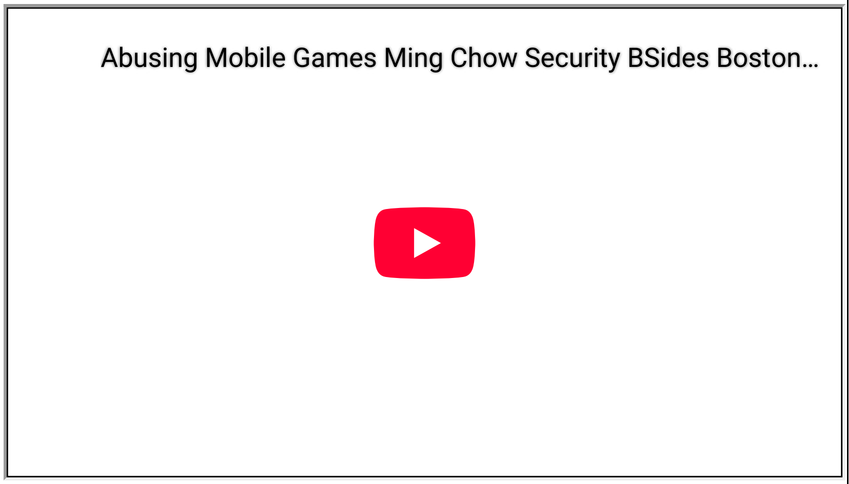
Schedule is subject to change.

Topics:

- Audio, Sound, and Music
- Storytelling
- Artificial Intelligence
- Game Testing
- Ethics, MMORPGs, and Securing Online Games
- Mobile Games
- Networking

Date	Agenda	Deliverables
Tuesday, September 8th	Course Introduction	• Lab 1: Course Roster. PLEASE COMPLETE ASAP! This lab is worth 1 point. • Please sign up for our Piazza group • Semester Personal Engagement Project

		• Sign up for the Fall 2015 Reverse Career Fair (not mandatory)
Thursday, September 10th	• Brief History of Video Games; Game Design Principles • Read: DEF CON: Why Conference Harassment Matters • Read: Sexual Harassment at DefCon (and Other Hacker Cons) by Bruce Schneier • Read: DEF CON Conference Code of Conduct	• The One Button Game Design Document (GDD)
Tuesday, September 15th	• Game Development Methodologies • Working in a Team; Agile; Git • Play: Zork I - The Great Underground Empire (by Infocom) • Read: Down From the Top of Its Game: The Story of Infocom, Inc. (MIT)	• Project 1 Assigned • Teams • Career Panel Hosted by Tufts CS from 7:30 - 9 PM in Cabot Auditorium. You will receive 2 points for attending this event. Attendance will be taken.
Thursday, September 17th	• Working in a Team; Agile; Git • Revision Control with Git (COMP 20 notes) • GitHub's Git Cheat Sheet • Reading: A Successful Git Branching Model	To Do Before Next Class: Complete the Phaser tutorial a http://phaser.io/tutorials/making your-first-phaser-game/index. Come to the next class with questions to ask. Please complete it in a git repository using the skills we've discusse in class, post this git repository on github, and send us a link via email (arthur.berman@tufts.edu).
Tuesday, September 22nd	• JavaScript • Q: Why Not Python and PyGame and Why Not Unity (for now)? • Phaser.io	
Thursday, September 24th		Lab Due 10/1
Tuesday, September 29th	Work Day	• Tufts Career Fair on Wednesday, September 30th from 11:30 AM - 2:30 PM at Gantcher • Tufts CS Reverse Career Fair on Wednesday, September 30th from 2:45 - 4 PM in Halligan 102
Thursday, October 1st	Sprites, Animation, Physics	
Tuesday, October 6th	Unity	
Thursday, October 8th	Project 1 Demo	
Tuesday, October 13th		

Thursday, October 15th		
Tuesday, October 20th	Project 1 Due	
Thursday, October 22nd		- Tufts Hackathon: Friday, October 23rd - Saturday, October 24th
Tuesday, October 27th	- Feedback on Game Project 1 - Game Testing - Personal Engagement Projects	Project 2 Assigned
Thursday, October 29th		
Tuesday, November 3rd		
Thursday, November 5th		
Thursday, November 12th		
Tuesday, November 17th	- Ethics, MMORPGs, and Securing Online Games - Abusing Mobile Games (my presentation at the BSides Boston Security Conference 2014) - Watch: 	IEEE Security & Privacy Securing Online Games Readings. Due on Tuesday, November 24th in class
Thursday, November 19th		
Tuesday, November 24th		
Tuesday, December 1st		
Thursday, December 3rd		

Tuesday, December 8th		
Thursday, December 10th		

Grading

- Group Project 1: 20%
- Group Project 2: 55%
- Labs and exercises: 15%
- Class participation: 10%
- Personal Engagement Project: 10%

Course Policies

Assignment Late Policy

Assignments (labs, homework) that is submitted electronically (most homework) are due at 11:59 PM on a Tuesday or Thursday. We will grant an automatic extension of ten minutes at no cost to you. If you plan on submitting your work at midnight or at six, you will have nine minutes for last-minute changes.

An assignment is expected to be submitted on time. However, we recognize that the exigencies of college life occasionally interfere with on-time submission. If you have difficulty getting the assignment in on time, you have two options:

1. For ordinary difficulties, each student is automatically issued three (3) "extension tokens." By expending an extension token, you can get an automatic 24-hour extension on all deadlines associated with a single assignment. To use an extension token, you must e-mail me at **`mchow@cs.tufts.edu`**. Thus must be sent before the assignment is due. At most two extension tokens may be expended on any single assignment. When you are out of tokens, late assignments will no longer be accepted: it will be returned ungraded, and you will receive no credit for the work.
2. If a serious illness affects your ability to complete the assignment on time, your first step is to report the illness using the "Illness Notification Form" that is available in WebCenter for Students. We will make suitable arrangements. For extraordinary difficulties, such as bereavement, family emergencies, or other extraordinary unpleasant events, your first step should be to make contact with your associate dean for undergraduate education. You must take this step before the assignment is due. Ask your dean to drop me an email or give me a call, and we will make special arrangements that are suited to your circumstances.

Please understand that extension tokens are meant to be used. That is, you will not receive any special bonus at the end of the course if you do not use any of your extension tokens.

Labs

A lab is due one week from the day it is assigned.

Solutions to Assignments and Examinations

Solutions to assignments and examinations will not be posted for this course.

COMP 150-SEN

Software Engineering Foundations, Spring 2019

Staff

Name	Office	E-mail	Office Hours (also available by appointment)
Jeff Foster	211 Halligan	jfoster@cs.tufts.edu	MW 1:15-2:00pm
Tomoki Shibata	Wed: Halligan extension 007; Fri: Halligan 209	tshibata@cs.tufts.edu	W 4:15-5:15pm, F 9:30-10:30am

Information

Location	Bromfield-Pearson Room 002
Time	MW 3:00-4:15
Midterm	March 13, in class
Final	May 3, 3:30-5:30pm
Textbooks	There are no required textbooks

Description

The scale of modern software systems is truly amazing. We regularly use software that is hundreds-of-thousands to millions of lines of code, and that software, while certainly not bug-free, mostly does what it is supposed to. Achieving this kind of scale has been the result of steady progress in *software engineering* over the last several decades.

In this class, we will study the foundations of software engineering, focusing on the core principles and ideas that enable us to build large-scale software systems. Our focus will be on the coding side of software engineering. Equally important, but deferred to a different course, is the people/team side of software engineering. We will explore ideas such as abstraction, modularity, architecture, specification, testing, and debugging, among others.

The course will be conducted in Java, and will include a short introduction to Java at the beginning. During the course, students will complete a number of programming assignments, and the course will also include some reading assignments.

Prerequities: COMP 40, graduate standing, or instructor consent. If you are not sure whether you meet the prerequisites for the course, please contact the instructor.

Schedule

Week	Monday	Wednesday
Jan 14	*(No class)*	*No class*
Jan 21	*MLK Day* No class	Introduction Java Reading Assignment #1
Jan 28	Java Project 1	Java LinkedList.java List.java

Feb 4	Java ArrayList.java	Java
Feb 11	Design Patterns Reading Assignment #2	Design Patterns Project 2
Feb 18	*President's Day* No class	(Feb 20) Modularity Discussion Design Patterns (Feb 21) Design Patterns ooc.c Reflection
Feb 25	Software Architecture Reading Assignment #3	Testing
Mar 4	Fuzz Testing Discussion Testing	Testing
Mar 11	Midterm Review	Midterm
Mar 18	*Spring Recess* No class	*Spring Recess* No class
Mar 25	Refactoring Project 3	Program Verification
Apr 1	*Guest Lecture:* Milod Kazerounian, Program Verification (cont'd)	*Guest Lecture:* Diogenes Nunez, Garbage Collection Reading Assignment #4
Apr 8	Security java-sec-example.tar.gz	No Silver Bullet discussion Project 4
Apr 15	*Patriots' Day* No class	Debugging How Failures Come to Be The Scientific Method Reproducing Problems
Apr 22	Simplifying Problems Fixing the Defect Reading Assignment #5	ACM Code of Ethics
Apr 29	What Makes a Great Software Engineer Final Exam Review	*(No class)*

Reading Assignments

1. (Due Jan 30) Tony Hoare, Null References: The Billion Dollar Mistake
   - It's okay to read the show notes instead of watch the video.
   - Q1: What was the most interesting thing you learned from the presentation?
   - Q2: Do you think null pointers really were a billion dollar mistake?
2. (Due Feb 20) D.L. Parnas, On the Criteria To Be Used in Decomposing Systems into Modules
   - Q1: Briefly describe one important lesson about modularity that the paper describes and that you think is still relevant today.
   - Q2: Computing has advanced significantly since this paper was written. Briefly describe one challenge (not necessarily with modularity) or perspective that, while realistic in 1972, does not apply today.
3. (Due Mar 4) Miller, Fredriksen, and So, An Empirical Study of the Reliability of UNIX Utilities
   - Q1: Briefly describe one strength and one weakness of fuzz testing.
   - Q2: Two modern fuzz testers are american fuzzy lop and libfuzzer. Pick one, look through the web page(s) for it, and briefly describe one way it improves on the original fuzz testing work of Miller et al.
4. (Due Apr 10) Brooks, No Silver Bullet: Essence and Accidents of Software Engineering
   - Q1: Briefly discuss one contrast Brooks draws between Software Engineering and other areas or science or engineering.
   - Q2: Brooks discusses several "Hopes for the silver", but his comments are about the state of the art in 1987. Describe one "hope for the silver" that you see today, but that Brooks did not see back then. Or, if you are a pessimist, explain why there is no hope.
5. (Due Apr 29) Li, Ko, and Zhu, What makes a great software engineer?
   - Q1: What was the most surprising characteristic to you of a great software engineering, as described by the paper?
   - Q2: What are some things you might want to do on your own, after this course ends, to become a great software engineer?

Projects

- Project 1 - Java ADTs and the Adapter Pattern
- Project 2 - Design Patterns in Java
- Project 3 - A Unit Testing Framework
- Project 4 - Java on Rails

Exams

- Sample questions for the midterm (solutions)
- Midterm sample solutions
- Final sample solutions

Syllabus

Syllabus subject to change until the start of the semester

Prerequisites

COMP 40, graduate standing, or instructor consent. If you are not sure whether you meet the prerequisites for the course, please contact the instructor.

List of Topics (Tentative)

Below is an approximate list of topics for class. The exact topics will be determined based on the pace of the class.

- Java programming, including classes, objects, inheritance, interfaces, delegation, Java generics, and the Java runtime
- Abstract data types, modularity, information hiding
- Design patterns, including for concurrency
- Software architecture
- Program specification and verification
- Object-oriented refactoring
- Testing
- Debugging
- Program synthesis
- Special topics (TBA)

Office Hours and Web Forum

Office hours for the instructional staff will be posted on the course web page a few days into the semester.

While we will provide assistance with assignments during office hours, you are responsible for developing and debugging your own programs. Do not rely on the instructional staff to make your project work.

Important announcements will be made in class or on Piazza. Please make it a habit to check Piazza daily, and/or sign up to receive email when updates are posted to Piazza. You may also use the class web forum to ask general questions of interest to the class as a whole, e.g., administrative issues or project clarification questions. Please do not post any information that would violate the University's Academic Integrity Policy.

Grading

You are responsible for all material discussed in class and posted on the class web page, including announcements, deadlines, policies, etc. Your final course grade will be determined according to the following percentages:

Projects/homework	50%
Readings	9%
Midterm	20%
Final	20%
Meet your professor	1%

Any request for reconsideration of any grading on coursework **must** be submitted within **one week** of when it is returned. Exam regrading requests must be made in writing. Any coursework submitted for reconsideration may be regraded in its entirety, which could result in a lower score if warranted.

Final course grades will be curved as necessary, based on each student's total numeric score for all coursework at the end of the semester. **Important:** Completing the programming assignments is an essential part of the course. Therefore, **we may fail any student who does not make a good-faith attempt on all course projects**, regardless of the student's performance or scores on the other coursework.

Programming Projects

Projects must be submitted electronically following the instructions given in class. Projects **may not** be submitted by any other means (e.g., please do not email your projects to us). It is **your responsibility** to test your program and verify that it works properly before submitting. All projects are due at 11:59pm on the day indicated on the project assignment, according to the submission server's internal clock.

Projects may be submitted up to 24 hours late for a 10% penalty. For example, a project that would earn 90 points for an on-time submission will earn 81 (which is 90 times 0.90) if submitted late. Note that your project score as it appears on the project submission server will not include any late penalties. Any penalties will be incorporated into the final project grade on the grade server.

Exam Scheduling

The class includes a midterm and a final exam. **Tentative** dates for the exams will be posted on the class web site. The exact dates will be confirmed later. We will let you know the exact dates well in advance.

Academic Resources

The Academic Resource Center offers a range of services for students.

Accommodations for Students with Disabilities

Tufts University values the diversity of our students, staff, and faculty, recognizing the important contribution each student makes to our unique community. Tufts is committed to providing equal access and support to all qualified students through the provision of reasonable accommodations so that each student may fully participate in the Tufts experience. If you have a disability that requires reasonable accommodations, please contact the Student Accessibility Services office to make an appointment with an SAS representative to determine appropriate accommodations. Please be aware that accommodations cannot be enacted retroactively, making timeliness a critical aspect for their provision.

Please also contact the instructor to discuss any necessary accommodations.

Excused Absences

You are expected to attend class regularly, complete course assignments on time, and take exams at the scheduled times. If you are unable to fulfill these requirements due to absence for a good reason, the instructor will excuse the absence and provide accommodation. Events that justify an excused absence include:

- Religious observances
- Mandatory military obligation
- Illness of the student or illness of an immediate family member
- Participation in university activities at the request of university authorities
- Compelling circumstances beyond the student's control (e.g., death in the family, required court appearance, etc.)

It is **your responsibility** to inform the instructor **in advance** of intended religious observances. Notice must be provided **immediately** upon an exam date being announced or confirmed for an absence to be excused.

The policies for excused absences **do not** apply to project assignments. Projects will be assigned with sufficient time to allow students to carry out the work even with other responsibilities. In cases of extremely serious documented illness of lengthy duration or other protracted, severe emergency situations, the instructor may consider extensions depending on the specific cirucmstances.

Absences stemming from job interviews, traffic or transportation problems, personal travel, and similar will not be excused.

Academic Integrity

The university's Academic Integrity Policy will be strictly enforced.

Unless otherwise specified, programming projects are to be written **individually**. Therefore, cooperation or use of unauthorized materials on projects is a violation of the Academic Integrity Policy. **Project solutions may not be posted online.** Any evidence of this, or of unacceptable use of computer accounts, use of unauthorized materials or cooperation on exams or quizzes, or other possible academic integrity violations will be reported.

For learning the course concepts, students are welcome to study together or to receive help from anyone else. You may discuss with others the project requirements, the features of the programming languages used, what was discussed in class and in the class web forum, and general syntax errors. Examples of questions that would be allowed are "Does a Java class definition end in a semicolon?" or "What does a 'class not found' error indicate?", because they convey no information about the contents of a project.

When it comes to actually writing a project assignment, other than help from the instructional staff a project must solely and entirely be your own work. Working with another student or individual, or using anyone else's work in any way except as noted in this paragraph, is a violation of the Academic Integrity Policy. You may not discuss design of any part of a project with anyone except the instructor or teaching assistants. Examples of questions you may not ask others might be "How did you implement this part of the project?" or "Please look at my code and help me find my stupid syntax error!". You may not use any disallowed source of information in creating either their project design or code. When writing projects you are free to use ideas or short fragments of code from published textbooks or publicly available information, but the specific source must be cited in a comment in the relevant section of the program.

Violations of the Code of Academic Integrity may include, but are not limited to:

1. Failing to do all or any of the work on a project by yourself, other than assistance from the instructional staff.
2. Using any ideas or any part of another person's project, or copying any other individual's work in any way.
3. Giving any parts or ideas from your project, including test data, to another student.
4. Allowing any other students access to your program on any computer system.
5. Transferring any part of a project to or from another student or individual by any means, electronic or otherwise.

If you have any question about a particular situation or source then consult with the instructor in advance. Should you have difficulty with a programming assignment you should **see the instructional staff in office hours**, and not solicit help from anyone else in violation of these rules.

Right to Change Information

Although every effort has been made to be complete and accurate, unforeseen circumstances arising during the semester could require the adjustment of any material given here. Consequently, given due notice to students, the instructor reserves the right to change any information on this syllabus or in other course materials.

Resources

- Java Basics
  - Java Development Kit (JDK) Download
  - JDK 11 Documentation
  - JDK 11 API
- Java Textbooks (read via Safari Books Online)
  - Head First Java
  - The Java Programming Language
  - Effective Java
  - Java Cookbook
- Related Papers
  - Garlan and Shaw, An Introduction to Software Architecture
  - Klees et al., Evaluating Fuzz Testing
- Other
  - Explaning Code using ASCII Art
  - Why Programs Fail (Slides)

COMP 260

Advanced Algorithms

Spring 2025

Instructor: Lenore Cowen
Joyce Cummings Center; 627-5134; cowen AT cs.tufts.edu ;
Office Hours: Typically 2-3pm on Monday and Wednesday
Lectures: Mondays/Wednesdays 3pm-4:15pm in JCC 302

See the private page here for the code to add yourself to Gradescope to submit your homework assignments.

Description: If you loved your algorithms class and can't wait for more, this is the class for you. In this pleasant and fun class, we will look at some more modern algorithms, some beautiful algorithms gems, and some areas of current research in algorithms. Topics will include using randomness in the design and analysis of algorithms, approximation algorithms, and online algorithms.

The website for this class is at http://www.cs.tufts.edu/comp/260

Prerequisites: Comp 160 or permission of the instructor.

There is no text for this course.

The course was last taught in 2023.

Here is a Template for scribe notes with instructions for first-time Latex users We can also supply this as an overleaf project-- just ask.

You can find the scribe notes for the class here: http://www.cs.tufts.edu/comp/260/private/

- Week 0/1: January 15 and 22 Review of P and NP, Bipartite perfect matching and Stable Matching
- Week 2: January 27 and 29 Approximation algorithms for Metric TSP , HW 1 due
- Week 3: Feb 3 and Feb 5 Knapsack I and II
- Week 4: Feb 10 and 12 Ford-Fulkerson Algorithm for Max Flow/Min Cut; HW1 due
- Week 5: Feb 19 and 20 Intro to Randomaized Algorithms, Min Cut, MaxCut
- Week 6: Feb 24 and 26 k-Center. Begin scheduling
- Week 7: March 3 and March 5 Set Cover, HW2 due
- Week 8: March 10 and March 12:
- No classes: March 17 and March 19: Spring break
- Week 9: March 31 and April 2: Intro to Online Algorithms
- Week 10: April 7 and April 9: Finish Online lecture and k-Center
- Week 11: April 14 and April 16: Max Cut Revisited and Approx 3 Coloring

- Week 12: April 21 and April 23: Patriot's Day and TBA
- Week 13: April 28 Last Class

Homework Assignments: The code to add yourself to Gradecope for this class to submit electronically is handed out in class or available on the private portion of this website.

HW1 (due Thurs, Jan 30 at 10pm) and here is the Latex source

HW2 (due Thurs, March 6 at 10pm) and here is the Latex source where you will also need This figure to compile the latex source.

The Bleeding Edge:

In this section we link to current research papers that are related to the topics of the lectures of this course.

- Lecture 1: More on stable marriage. See recent work on Generalized Median Stable Matchings of Christine Cheng's on Her webpage
- Lecture 3: While there has been no improvement on Christofides algorithm for metric TSP in terms of approximation factor, there has been more recent approximation algorithms for metric TSP s-t path (i.e. you don't close the cycle) and some very recent work on nearly achieving the same approximation guarantee of Christofides algorithm with a better running time:
- Lecture 4: A PTAS for the Multiple Knapsack problem (a generalization of Knapsak) was found by Chekuri and Khanna in 2006: C. Chekuri, S. Khanna, "A polynomial time approximation scheme for the multiple knapsack problem", SIAM Journal on Computing 35(3), 713-728.
-

COMP170 - Spring 2025

HOME SCHEDULE POLICY RESOURCES

ow you will find our **proposed** schedule for the semester. Topics and assignments are subject to change, and may lated as the semester progresses.

eek 1: 1/12/2025

Wednesday	**Recitation 0:** handout | solution
Thursday	**Lecture:** Introduction Reading: Gödel's Proof, Chapters 1-4 **HW0 Out:** pdf

eek 2: 1/19/2025

Monday	No School
Tuesday	**Lecture:** Paradoxes Reading: Gödel's Proof, Chapter 5
Wednesday	Fake Monday
Thursday	**Lecture:** Gödel Reading: Gödel's Proof, Chapter 7

eek 3: 1/26/2025

Tuesday	**Lecture:** Turing Reading: The Annotated Turing, Chapter 5 **HW0 Due** by the start of class **HW1 Out:** pdf | tex
Wednesday	**Recitation 1:** handout | solution
Thursday	**Lecture:** The Universal Turing Machine Reading: The Annotated Turing, Chapter 9 **QUIZ 1**

eek 4: 2/2/2025

Tuesday	**Lecture:** The Un-Halting Problem Reading: The Annotated Turing, Chapter 9 **HW1 Due** by the start of class **HW2 Out:** pdf | tex
Wednesday	**Recitation 2:** handout | solution

Thursday	**Lecture:** Deciders Reading: The Annotated Turing, Chapter 10

eek 5: 2/9/2025

Tuesday	**Lecture:** The Halting Problem Reading: Sipser, Chapter 4 **HW2 Due** by the start of class **HW3 Out:** pdf | tex
Wednesday	**Recitation 3:** handout | solution
Thursday	**Lecture:** Rice's Theorem Reading: Sipser, Chapter 5 **QUIZ 2**

eek 6: 2/16/2025

Monday	No School
Tuesday	**Lecture:** Mapping Reductions Reading: Sipser, Chapter 5.3 **HW3 Due** by the start of class **HW4 Out:** pdf | tex
Wednesday	**Recitation 4:** handout | solution
Thursday	Fake Monday

eek 7: 2/23/2025

Tuesday	**Lecture:** Finite State Automata Reading: Sipser, Chapter 1.1 **HW4 Due** by the start of class **HW5 Out:** pdf | tex
Wednesday	**Recitation 5:** handout | solution
Thursday	**Lecture:** The Pumping Lemma Reading: Sipser, Chapter 1.4 **QUIZ 3**

eek 8: 3/2/2025

Tuesday	**Lecture:** Nondeterministic Finite State Automata Reading: Sipser, Chapter 1.2 **HW5 Due** by the start of class **HW6 Out**
Wednesday	**Recitation 6:** handout | solution

Thursday	**Lecture:** Context Free Grammars Reading: Sipser, Chapter 2

eek 9: 3/9/2025

Tuesday	**Lecture:** CFGs Continued **HW6 Due** by the start of class **HW7 Out**
Wednesday	**Recitation 7:** handout | solution
Thursday	**Lecture:** The Onion **QUIZ 4**

eek 10: 3/16/2025

Monday	No School
Tuesday	No School
Wednesday	No School
Thursday	No School
Friday	No School

eek 11: 3/23/2025

Tuesday	**Lecture:** P vs. NP Reading: Sipser, Chapter 7.1-7.3 **HW7 Due** by the start of class **HW8 Out**
Wednesday	**Recitation 8:** handout | solution
Thursday	**Lecture:** The Cook Levin Theorem Reading: Sipser, Chapter 7.4

eek 12: 3/30/2025

Tuesday	**Lecture:** Graphs Reading: Sipser, Chapter 7.5 **HW8 Due** by the start of class **HW9 Out**
Wednesday	**Recitation 9:** handout | solution
Thursday	**Lecture:** Closure **QUIZ 5**

eek 13: 4/6/2025

Tuesday	**Lecture:** Subset Sum **HW9 Due** by the start of class **HW10 Out**
Wednesday	**Recitation 10:** handout | solution
Thursday	**Lecture:** TBD

eek 14: 4/13/2025

Tuesday	**Lecture:** P-Space Reading: Sipser, Chapter 8.1 **Extra Credit Homework Out** **HW10 Due** by the start of class
Wednesday	
Thursday	**Lecture:** Shor's Algorithm Reading: Algorithms (Dasgupta, Papadimitriou, Vazirani), Chapter 10 **QUIZ 6**

eek 15: 4/20/2025

Monday	No School
Tuesday	**Lecture:** TBD **Extra Credit Homework Due**
Wednesday	**Recitation 11:** handout | solution
Thursday	**Lecture:** Wrap Up

eek 16: 4/27/2025

Tuesday	
Wednesday	
Thursday	

eek 17: 5/4/2025

Monday	Final Exam: 3:30-5:30pm
Tuesday	
Wednesday	
Thursday	

```
Field1,Field2,Field5_text,Field6_text,Field7,Field8_text
ES 2-01,"MWF 10:30-11:45
Anderson Hall 112",Introduction to Computing in
Engineering,E. Danahy,E. Danahy,
CS 4-01,,Teaching Computer Science,M. Monroe,M. Monroe,
CS 5-01,"T 4:30-6:30p
Eaton Hall 201",Preparing for Career Success,,K. Donoghue,
CS 10-01,"MW 1:30-2:45
Room To Be Announced",Computer Science for All,,E. Cornwall,
CS 11-M1,"T 5:30p-7:00p
Online (synchronous)",Introduction to Computer
Science,,P. Biswas,CS 11-M1
CS 11-01,"TR 10:30-11:45
Room To Be Announced",Introduction to Computer
Science,,E. Cornwall,CS 11-01
CS 11-02,"TR 12:00-1:15
Room To Be Announced",Introduction to Computer
Science,,E. Cornwall,CS 11-02
CS 14-01,,Emerging Scholars in Computer
Science,R. Townsend,R. Townsend,
CS 15-M1,"R 5:30p-7:00p
Online (synchronous)",Data Structures,,C. Magnano,CS 15-M1
CS 15-01,"MW 10:30-11:45
Room To Be Announced",Data Structures,,M. Kazerounian,CS 15-01
CS 15-02,"MW 3:00-4:15
Room To Be Announced",Data Structures,,M. Kazerounian,CS 15-02
CS 20-01,"TR 12:00-1:15
Room To Be Announced",Web Programming,,L. DiOrio,CS 20-01
CS 21-01,"TR 4:30-5:45p
Room To Be Announced",Concurrent Programming,M. Sheldon,M. Sheldon,
CS 23-01,"T 6:00p-9:00p
Room To Be Announced",Game Design,,J. Wiser,CS 23-01
CS 27-01,"TR 10:30-11:45
Room To Be Announced",How Systems Fail,,M. Kazerounian,
CS 28-01,"TR 1:30-2:45
Room To Be Announced",Cyber Security and Cyber
Warfare,M. Chow,"M. Chow,  J. Taliaferro",
CS 30-M1,"W 6:00p-7:30p
Online (synchronous)",Programming for Data Science,A. Couch,A. Couch,
CS 40-01,"TR 1:30-2:45
Room To Be Announced",Machine Structure & Assembly-Language
Programming,N. Mendelsohn,"N. Mendelsohn,  M. Sheldon",CS 40-01
CS 61-M1,"T 7:00p-8:30p
Online (synchronous)",Discrete Mathematics,,M. Jahn,CS 61-M1
CS 61-01,"MWF 9:30-10:20
Room To Be Announced",Discrete Mathematics,,M. Allen,CS 61-01
CS 61-02,"MWF 10:30-11:20
Room To Be Announced",Discrete Mathematics,,M. Allen,CS 61-02
CS 93-01,,Directed Study,,Members of the Department,
CS 98-01,"F 12:00-2:45
```

```
Room To Be Announced",Senior Capstone Project II,,D. Lillethun,
CS 99-01,,,M. Chow,M. Chow,
CS 105-01,"MW 3:00-4:15
Room To Be Announced",Programming
Languages,R. Townsend,R. Townsend,CS 105-01
CS 111-M1,"T 7:00p-8:30p
Online (synchronous)",Operating Systems,,P. Biswas,
CS 111-01,"TR 4:30-5:45p
Joyce Cummings Center 170",Operating Systems,,D. Lillethun,
CS 114-01,"TR 12:00-1:15
Room To Be Announced",Network Security,D. Votipka,D. Votipka,
CS 115-M1,"T 7:00p-8:30p
Online (synchronous)",Database Systems,,C. Doucette,CS 115-M1
CS 115-01,"M 7:00p-8:30p
Room To Be Announced",Database Systems,,C. Doucette,CS 115-01
CS 116-M1,"W 5:30p-7:00p
Online (synchronous)",Introduction to Security,M. Chow,M. Chow,CS 116-
M1
CS 116-01,"TR 4:30-5:45p
Room To Be Announced",Introduction to
Security,M. Chow,M. Chow,CS 116-01
CS 119-M1,"M 5:30p-7:00p
Online (synchronous)",Big Data,,J. Singh,CS 119-M1
CS 119-01,"MW 10:30-11:45
Room To Be Announced",Big Data,,J. Singh,CS 119-01
CS 120-M1,"T 5:30p-7:00p
Online (synchronous)",Web Programming and Engineering,,L. DiOrio,
CS 121-M1,"W 7:30p-9:00p
Online (synchronous)",Software Engineering,,Staff,CS 121-M1
CS 122-01,"MW 1:30-2:45
Room To Be Announced",Parallel Computing,,J. Grodstein,
CS 125-01,"MW 1:30-2:45
Joyce Cummings Center 265",Numerical Analysis,,S. Lee,
CS 131-M1,"M 7:30p-9:00p
Online (synchronous)",Artificial Intelligence,,F. Santini,CS 131-M1
CS 131-01,"MW 6:00p-7:15p
Room To Be Announced",Artificial Intelligence,,F. Santini,CS 131-01
CS 132-01,"MW 4:30-5:45p
Room To Be Announced",Computer Vision,,R. Shilkrot,
CS 135-M1,"F 5:30p-7:00p
Online (synchronous)",Introduction to Machine
Learning,,C. Magnano,CS 135-M1
CS 135-01,"TR 12:00-1:15
Room To Be Announced",Introduction to Machine
Learning,,M. Hughes,CS 135-01
CS 138-M1,"R 7:00p-8:30p
Online (synchronous)",Reinforcement Learning,,Y. Shukla,
CS 138-01,"MW 3:00-4:15
Room To Be Announced",Reinforcement Learning,J. Sinapov,J. Sinapov,
CS 142-01,"MW 4:30-5:45p
```

```
Room To Be Announced",Network Science,L. Cowen,L. Cowen,
CS 144-01,"T 1:30-4:00
Room To Be Announced",Iterative Methods in Machine
Learning,U. Khan,U. Khan,
CS 150-01,"TR 9:00-10:15
Room To Be Announced",HCI for Disability,E. Short,E. Short,
CS 150-02,"MW 10:30-11:45
Room To Be Announced",,R. Townsend,R. Townsend,
CS 150-04,"MW 10:30-11:45
Room To Be Announced",,,F. Dogar,
CS 150-05,"TF 12:00-1:15
Room To Be Announced",,R. Mirsky,R. Mirsky,
CS 150-06,,MITRE eCTF,M. Chow,"S. Bell,  M. Chow",
CS 150-07,"T 5:15p-8:15p
Room To Be Announced",Entrepreneurship for Computer Scientists,,,
CS 150-08,"TR 12:00-1:15
Room To Be Announced",Topics in Computational
Complexity,,V. Podolskii,
CS 151-01,"TR 10:30-11:45
Room To Be Announced","Addressing Cyber Threats,
Vulnerabilities",,L. Weissinger,
CS 151-02,"M 5:30p-7:00p
Joyce Cummings Center 402",Cybersecurity Clinic,M. Chow,M. Chow,
CS 151-03,"TR 4:30-5:45p
Room To Be Announced","Privacy, Security, and Data",,J. Bater,
CS 151-04,"TR 1:30-2:45
Room To Be Announced",Anonymous Communications Theory,,M. Ando,
CS 151-05,"MW 1:30-2:45
Room To Be Announced",Quantum Computer Science,,S. Mehraban,
CS 151-06,"MW 4:30-5:45p
Room To Be Announced",Debugging Cloud Computing,,R. Sambasivan,
CS 151-07,"TR 9:00-10:15
Room To Be Announced",Sustainable Computing,,H. Sundar,
CS 160-M1,"W 7:00p-9:00p
Online (synchronous)",Algorithms,A. Winslow,A. Winslow,CS 160-M1
CS 160-01,"MW 9:00-10:15
Room To Be Announced",Algorithms,,K. Edwards,CS 160-01
CS 167-01,"TR 10:30-11:45
Room To Be Announced",Computational
Biology,D. Slonim,D. Slonim,CS 167-01
CS 170-M1,"R 5:30p-7:00p
Online (synchronous)",Computation Theory,A. Winslow,A. Winslow,CS 170-
M1
CS 170-01,"TR 1:30-2:45
Room To Be Announced",Computation Theory,M. Monroe,M. Monroe,CS 170-01
CS 171-01,"MW 1:30-2:45
Room To Be Announced",Human-Computer
Interaction,R. Jacob,R. Jacob,CS 171-01
CS 178-01,"TR 1:30-2:45
Room To Be Announced",Visual Analytics,R. Chang,R. Chang,CS 178-01
```

```
CS 183-01,"MTWRF 12:00a-12:00a
Room To Be Announced",Privacy in the Digital Age,,,
CS 185-01,"R 6:30p-9:00p
Room To Be Announced",Computing for Developing Regions,,F. Dogar,
CS 191-01,,Research,,Members of the Department,
CS 193-01,,Directed Study,,Members of the Department,
CS 193-02,,Geometric Folding Algorithms,D. Souvaine,D. Souvaine,
CS 193-03,,MS CoreComp,,K. Edwards,
CS 197-01,,Honors Thesis,,Members of the Department,
CS 203-01,"TR 10:30-11:45
Room To Be Announced",How Systems Fail,,M. Kazerounian,
CS 239-01,"MW 9:00-10:15
Room To Be Announced","Ethics for AI, Robotics, and Human Robot
Interaction",V. Sarathy,V. Sarathy,
CS 260-1,"MW 3:00-4:15
Room To Be Announced",Advanced Algorithms,L. Cowen,L. Cowen,CS 260-1
CS 288-M1,"R 6:00p-7:00p
Online (synchronous)",Master of Science Capstone Project I,,M. Allen,
CS 289-M1,"R 7:00p-8:00p
Online (synchronous)",Master of Science Capstone Project II,,M. Allen,
CS 291-01,"seminar: R 3:00-4:15
Joyce Cummings Center 270",Seminar in Computer
Science,D. Souvaine,"J. Redmond,  D. Souvaine",
CS 291-02,"seminar: F 1:30-2:30
Robinson Hall 253",Seminar in Computer Science,J. Sinapov,J. Sinapov,
CS 293-01,,Graduate Special Topics / Master's
Project,,Members of the Department,
CS 296-01,,Master's Thesis,,Members of the Department,
CS 297-01,,Graduate Research,,Members of the Department,
CS 299-01,,Internship in Computer Science,M. Chow,M. Chow,
CS 401-PT,,,,Members of the Department,
CS 402-FT,,,,Members of the Department,
CS 404-01,,,,A. Abdulrazzaq,
CS 405-TA,,,,Members of the Department,
CS 406-RA,,,,Members of the Department,
CS 501-PT,,,,Members of the Department,
CS 502-FT,,,,Members of the Department,
DS 93-01,,Directed Study,,Members of the Department,
DS 98-01,"TR 1:30-2:45
Online (synchronous)",Senior Capstone Project in Data Science
II,A. Couch,A. Couch,
DS 143-01,"T 1:20-4:20
Bromfield-Pearson 005",Data Science for Sustainability,,D. Sunter,
DS 143-02,"M 6:00p-9:00p
Anderson Hall 108",Data Science for Sustainability,,D. Sunter,
DS 153-02,"MW 6:00p-7:15p
Eaton Hall 201",Computer Vision,,R. Shilkrot,DS 153-02
DS 153-03,"MW 10:30-11:45
Collaborative Learning and Innovation Complex 401",,,F. Dogar,
DS 153-04,"TF 12:00-1:15
```

```
Tisch Library 316",,R. Mirsky,R. Mirsky,
DS 193-01,,Directed Study,,Members of the Department,
DS 288-M1,"T 7:00p-9:00p
Online (synchronous)",Master of Science Capstone Project I,,M. Allen,
DS 289-M1,"T 8:00p-9:00p
Online (synchronous)",Master of Science Capstone Project II,,M. Allen,
DS 293-01,,Masters Project in Data Science,,Members of the Department,
DS 299-01,,Internship In Data Science,A. Couch,A. Couch,
DS 404-01,,,,A. Abdulrazzaq,
```

CS 15 - Spring 2025

HOME SCHEDULE REFERENCE ADMIN

Welcome to CS 15!

CS 15 is all about building your programming toolkit. The tools are data structures, the means by which we store, organize, and access data. We will implement and utilize the key data structures and algorithms that every programmer must know, learn about the useful abstractions that they provide, and study the costs in time and space associated with them.

The structure and assignments of this course are based largely on prior versions of the course taught by Marty Allen, Mark Sheldon, Mati Korman, Chris Gregg, Ben Hescott, and Bruce Molay. Thank you!

Course Information

Course Policies: All students are expected to know and adhere to our course policies. Please take the time to read our admin page in full.

Prerequisites: CS 11 or consent of instructor.

Textbook: No purchase required. We will regularly post online readings from *Data Structures and Algorithm Analysis* by Clifford A. Shaffer.

Instructor: Milod Kazerounian
Email: `milod.kazerounian@tufts.edu`
Office: JCC 469
Office Hours: Tuesdays from 1-2pm in JCC 469, Thursdays from 2-3pm on zoom

Our graduate TA is Max Liu. Our teaching fellows are Arya Prasad, Hameedah Lawal, Kathy Quintanilla, and Liam Drew. We also have a horde of teaching assistants who will hold office hours in the third floor common space in the Joyce Cummings Center. Schedules for office hours will be regularly updated on our course forum, Piazza.

Lecture Section 1: (Mon/Wed) 10:30 – 11:45am
Location: Barnum Hall LL08
Final exam: May 2nd from 12-2pm

Lecture Section 2: (Mon/Wed) 3:00 – 4:15pm
Location: Robinson 253
Final exam: May 2nd from 3:30-5:30pm

For Spring 2025, both lectures will be automatically recorded by Tufts IT and posted on the CS 15 Canvas site under Echo 360. If you miss a lecture, you can find it there.

We do not use Canvas for anything else in CS 15!

Because all lectures are recorded, we are all on video with audio. The videos are for use by this class only and must not be downloaded or distributed.

CS 40 — Spring 2025

- Home
- Calendar
- Assignments
- Admin
- Resources

Announcements

This calendar represents the current plan, but it is subject to change. Please refresh frequently and look for announcements on Piazza.

All submissions are due by **11:59 pm (23:59)** at the end of the specified due date. Design documents should be submitted before **6 pm,** because submissions as of that time will be graded and have their feedback returned first. This means you can apply that feedback to the final submission! Design submissions made after 6 pm and before 11:59 pm will not be charged an extension token, but the feedback may not arrive in time to be incorporated into your final submission.

Read the specs carefully to understand the submission expectations for each assignment.

Mon	Tue	Wed	Thu	Fri
		15Jan Classes begin	16Jan Intro to CS 40 and Transitioning to C Read: C++ to C slides Self-study module: the transition from C++ to C (due Friday) CS 40 Admin & Policies Hanson: Ch. 1 & 2; Intro and First Section (titled "Interface") of Ch. 3-4, 7-9, 11; Section 8.2 Piazza HW 1: filesofpix	17Jan Lab 1: Rawness (HW1 Partner)

20Jan
Martin Luther King Day
	21Jan
	Abstraction, Modularity, Interfaces, and Pointers
	Read:
	Introduction to Exceptions
	C Idoms for CS 40
		22Jan *(Monday schedule)*
		Add deadline
			23Jan
			More on Abstract Data Types
			Read:
			UArray
			Void and Function Pointers
			Hanson's table interface, analyzed
			`cbr_example.c`
			`list_map_slide_2024.c`
			`list_map_slide_accum_2024.c`
				24Jan
				Lab 2: Unboxed arrays and 2D Arrays (HW2 Partner)

	Invariants		`list_map_stats.c` HW 1 design due (6 pm)	
27Jan	28Jan What is a Bit?	29Jan HW 1 due HW 2: iii	30Jan Building C Programs Read: B&O: Ch. 1 through 1.3; Sec. 1.7.4, 3.2.2 (use of objdump); Ch. 7 through 7.3 What happens when you build a program with gcc?	31Jan Lab 3: Problem-solving and black (HW2 Partner) Fall incompletes deadline
03Feb HW 2 design due (6 pm)	04Feb Process Memory Read: B&O: 1.7.1, 1.7.3. 9.1, 9.2 `memory.c`	05Feb	06Feb The Memory Hierarchy and Intro to Caching Read: Powerpoint with animations for slides B&O: Ch. 1.5, 1.6, Introduction to Ch. 6, 6.1.1 and 6.1.2 (just the first paragraph of each), 6.2-6.3	07Feb Lab 4: Work on HW2 (HW2 Partner)
10Feb HW 2 due HW 3: locality	11Feb More on Caching Read: Slides Powerpoint with animations B&O: The remainder of Ch. 6 Optional: A true story about a cache flex	12Feb	13Feb Object Oriented Programming	14Feb Lab 5: Striding through memory (HW3 Partner)
17Feb Presidents' Day HW 3 design due (6 pm)	18Feb Numbers and Machine Arithmetic Read: Integer slides Powerpoint with animations for Integer slides Floating point slides Floating point powerpoint with animations	19Feb Drop deadline	20Feb *(Monday schedule)* HW 3 due HW 4: arith Alan Turing proposes Turing test 20 Feb 1947	21Feb Lab 6: Diff (HW4 Partner)

	B&O: 2.2 – 2.4, Useful reference on numeric data (from Liam) fact.c fact.erl			
24Feb	25Feb Shifting, Masking and Endianness Read: Shifting and Masking slides Powerpoint with animations for Shifting and Masking slides Bit and Little Endianness slides Big and Little Endianness powerpoint with animations B&O: Start of chapter 2 – 2.1 endian.c	26Feb HW 4 design due (6 pm)	27Feb Introduction to Assembly Language Read: `hello_world_x86.s` B&O: 3.1-3.7 (warning: examples are not AMD64), 3.8-3.11 (skim), 3.13 This AMD64 Cheat Sheet from Brown is by far my favorite! For fun: IBM System 370 Reference Card	28Feb Lab 7: Bits and Shifts (No Partner)
03Mar	04Mar Assembly Continued Read: Assembler demos you can try Optional: Real programmers use machine code First Cray-1 shipped to LANL (1977) 	05Mar	06Mar Assembly Finale Read: Factorial Assembly Code HW 4 due HW 5: bomb	07Mar Lab 8: The GNU Debugger (HW5 Partner) Following are due on or happen on Saturday 08Mar: HW 4arith challenge Following are due on or happen on Sunday 09Mar: HW 4arith challenge due

10Mar	11Mar Midterm Review Read: Midterm Prep Refactoring and practice	12Mar	13Mar Midterm Read: Midterm solutions	14Mar Lab 9: Closing the loop on reverse engineering (HW5 Partner)
17Mar Spring break	18Mar Spring break	19Mar Spring break	20Mar Spring break	21Mar Spring break
24Mar Begin advising period	25Mar Introducing the CS 40 Universal Machine	26Mar	27Mar Implementing the Universal Machine HW 5 due HW 6: um	28Mar Lab 10: Unit testing for the UM (HW6 Partner)
31Mar	01Apr Performance Introduction Read: B&O: 1.9.1, Intro to Ch. 5, 5.7 (many details about how processor architecture affects performance) Be a CA (TA)!	02Apr HW 6 design due (6 pm) Withdraw deadline PF deadline	03Apr Performance Tools Read: B&O: Sec. 5.14 First cell phone call 3 Apr 1973	04Apr Lab 11: Profiling Tools (HW7 Partner)
07Apr IBM announces System 360 computer family 7 Apr 1964	08Apr Performance Tuning Read: B&O: Sec. 5.1, 5.4-5.6 inlining.c specialization.c	09Apr	10Apr The UM Macro Assembler Read: UM Macro Assembler Language Specification UMASM Lecture Notes HW 6 due HW 7: profiling End advising period	11Apr Lab 12: Extending the UM (No Partner) Begin fall registration
14Apr	15Apr Programming with UMASM	16Apr End fall registration First Lisp paper, by John	17Apr UMASM Programming Review and Advanced Examples Read: UMASM examples	18Apr Lab 13: Scripting/Q&A + HW 7 work

		McCarthy 16 Apr 1960		
21Apr Patriot's Day (Observed) HW 7 due HW 8: asmcoding	22Apr No lecture: work on HW	23Apr	24Apr CS 40 Wrap-up Final exam info Read: Final exam prep The UM in Python	25Apr Lab 14: Work on HW8 (HW8 Partner)
28Apr HW 8 due Classes end Grad student withdraw deadline	29Apr *Make up day: meet only if we've missed classes* Reading period	30Apr Reading period Claude Shannon's birthday 30 Apr 1916	01May Reading period	02May Begin final exams
05May Final exam *Tentatively 1530–1730 (i. e., 3:30–5:30pm)* Location: TBA	06May	07May	08May	09May End final exams
12May Degree candidates' grades due 0900 Begin Senior Week	13May	14May	15May	16May Remaining grades due 0900 End Senior Week Following are due on or happen on Sunday 18May: Commencement

CS 40 — Spring 2025

HOME CALENDAR ASSIGNMENTS ADMIN RESOURCES

Course Summary

Welcome to CS 40 – Machine Structure and Assembly Language Programming! CS 40 will give you two important experiences:

1. You will turn the corner from introductory programming to programming as a professional.
2. You will study in detail the structure of modern computers, and that in turn will add greatly to your understanding of how computer hardware, operating systems, compilers, language runtimes, and other tools work together to influence the performance of your programs.

These two goals are very closely related: all good computer professionals know how to apply deep knowledge of computing systems to their programming projects and to their research in computer science. Whether you are a newly confirmed CS major, someone who is considering a CS major but is unsure, or a student from another field who is interested in a deep look at some of the most important technologies of our time, CS 40 should give you an intense, realistic and (we hope!) exciting look at how modern computers work.

New CS 40 Students

Many students are eager to get a head start on the CS 40 material so that they can arrive as prepared as possible to the first day of class. Here is what you can do to prepare:

1. When CS 40 begins, you will be making an immediate transition from C++ to C. Be ready for this. Our resources page provides some great tutorials on this transition.
2. Read our admin page. This page is a comprehensive review of our course policies. You will be expected to adhere to these policies, so a good start is to know what they are. Pay particular attention to our course coding standands and our pair programming conduct guidelines.
3. Visit the course Piazza forum. (Your instructor will add you to the Piazza roster at the start of the term.)
4. Make sure that you have access to our course textbooks, which are listed below. In particular, the Hanson book describes the Swiss army knife of tools that will be provided to you for use throughout the semester.
5. Familiarize yourself with our Halligan Helper system. This is how students can request help from TAs during office hours.

Textbooks

The following two textbooks are required:

Textbook: *C Interfaces and Implementations* by David R. Hanson
ISBN: 9780201498417
Online access: via Tufts Library
There is also a PDF Quick Reference.

Textbook: *Computer Systems: A Programmer's Perspective*, 3rd edition by Bryant and O'Hallaron

ISBN: 9780134092669
Important: Make sure you avoid the "International" or "Global" version of this textbook, which is in paperback or the Amazon ebook. While substantially cheaper, many of the chapter practice problems have been rewritten by the publisher and contain many errors and inconsistencies.

Additional books that you may fine useful are listed on our resources page.

Schedule

Lectures: Tuesdays / Thursdays 1:30 – 2:45 PM EST
Location: SEC Robinson 253

Labs: Friday 10:30 – 11:45 AM, 12:00 – 1:15 PM, and 1:30 – 2:45 PM EST
Location: Joyce Cummings Center 235 and 240

Midterm Exam: See the course calendar
Final Exam: See the course calendar

Staff

Instructor: Mark Sheldon
Email: msheldon@cs.tufts.edu
Office hours: Will be posted at https://www.cs.tufts.edu/~msheldon about the second week of the semester.
Instructor: Noah Mendelsohn
Email: noah@cs.tufts.edu
Office hours: Will be posted at https://www.cs.tufts.edu/~noah/office_hours.php about the second week of the semester.

Our teaching fellows are Dan Bergen, Valerie Zhang, Sarah Svahn, and Julia Zelevinski. We also have an army of undergraduate teaching assistants who will hold office hours. Schedules for the undergraduate assistants will be posted and updated on Piazza.

Please refer to our admin page for guidelines on how to contact our course staff with questions and feedback.

Home

CS 61: Discrete Mathematics - Schedule

Main page • Syllabus

DATE	ACTIVITY	TOPIC	READINGS
WEEK 1			
Wed. Jan. 15	Lecture 1A	Intro to discrete mathematics	Syllabus
Fri. Jan. 17	Lecture 1B	Fundamental reasoning	Scheinerman (SCH), sections 1–4
WEEK 2			
Mon. Jan. 20		**No class (MLK day)**	
Wed. Jan. 22	Lecture 2A	Proofs	SCH, section 5
Fri. Jan. 24	Lecture 2B	Counterexamples; Boolean algebra	SCH, sections 6–7
WEEK 3			
Mon. Jan. 27	**Homework 1**	Submit on Gradescope by 11:59pm	
Mon. Jan. 27	Lecture 3A	More on Boolean algebra	SCH, section 7
Wed. Jan. 29	Lecture 3B	Lists; factorials	SCH, section 8
Fri. Jan. 31	Lecture 3C	Sets	SCH, sections 9–10
WEEK 4			
Mon. Feb. 03	**Homework 2**	Submit on Gradescope by 11:59pm	
Mon. Feb. 03	Lecture 4A	Quantifiers	SCH, section 11
Wed. Feb. 05	Lecture 4B	Set operations	SCH, section 12
Fri. Feb. 07	Lecture 4C	Combinatorial proofs	SCH, section 13
WEEK 5			
Mon. Feb. 10	**Homework 3**	Submit on Gradescope by 11:59pm	
Mon. Feb. 10	Lecture 5A	Sets	SCH, section 12
Wed. Feb. 12	Review day	Review for exam	
Fri. Feb. 14	**Midterm 01**		Exam is held during usual class meeting, 170 JCC
WEEK 6			
Mon. Feb. 17		**No class (Presidents' Day)**	
Wed. Feb. 19	Lecture 6A	Sets and combinatorial proofs	SCH, sections 12–13

Thu. Feb. 20 **(Tufts Monday)**	Lecture 6B	Relations	SCH, section 14
Fri. Feb. 21	Lecture 6C	Equivalence relations	SCH, section 15
WEEK 7			
Mon. Feb. 24	**Homework 4**	Submit on Gradescope by 11:59pm	
Mon. Feb. 24	Lecture 7A	Partitions	SCH, section 16
Wed. Feb. 26	Lecture 7B	Binomial coefficients	SCH, section 17
Fri. Feb. 28	Lecture 7C	Binomial coefficients, cont'd.	
WEEK 8			
Mon. Mar. 03	**Homework 5**	Submit on Gradescope by 11:59pm	
Mon. Mar. 03	Lecture 8A	Proofs by contradiction and contrapositive	SCH, section 20
Wed. Mar. 05	Lecture 8B	Proof by induction	SCH, section 22 (up to, not including, start of Strong Induction, p. 141)
Fri. Mar. 07	Lecture 8C	Strong induction	SCH, section 22 (141–145)
WEEK 9			
Mon. Mar. 10	**Homework 6**	Submit on Gradescope by 11:59pm	
Mon. Mar. 10	Lecture 9A	Functions	SCH, section 24
Wed. Mar. 12	Lecture 9B	Functions, cont'd.; the Pigeonhole Principle	SCH, section 25
Fri. Mar. 14	Lecture 9C	Function composition	SCH, section 26
WEEK 10			
Mon. Mar. 24	Lecture 10A	Big-O notation and functions	SCH, section 29 (up to page 207 only)
Wed. Mar. 26	Review day	Review for exam	
Fri. Mar. 28	**Midterm 02**		Exam is held during usual class meeting, 170 JCC
WEEK 11			
Mon. Mar. 31	**Homework 7**	Submit on Gradescope by 11:59pm	
Mon. Mar. 31	Lecture 11A		
Wed. Apr. 02	Lecture 11B		
Fri. Apr. 04	Lecture 11C		
WEEK 12			

Mon. Apr. 07	**Homework 8**	Submit on Gradescope by 11:59pm;	
Mon. Apr. 07	Lecture 12A		
Wed. Apr. 09	Lecture 12B		
Fri. Apr. 11	Lecture 12C		
WEEK 13			
Mon. Apr. 14	**Homework 9**	Submit on Gradescope by 11:59pm;	
Mon. Apr. 14	Lecture 13A		
Wed. Apr. 16	Lecture 13B		
Fri. Apr. 18	Lecture 13C		
WEEK 14			
Mon. Apr. 21	**Homework 10**		Submit on Gradescope by 11:59pm
Mon. Apr. 21	Lecture 14A		
Wed. Apr. 23	Review Day	Review for Exam	
Fri. Apr. 25	**Midterm 03**		
WEEK 15			
Mon. Apr. 28	Last day	Wrap-up and review	
FINALS WEEK			
Fri. May 02	**Final (section 02)**		Exam is held 12:00–2:00 PM, 170 JCC
Tue. May 06	**Final (section 01)**		Exam is held 7:00–9:00 PM, 170 JCC

CS 61: Discrete Mathematics - Syllabus

Main page • Schedule

Description and Objective: An introduction to foundational mathematical concepts and techniques: Sets, relations and functions, logic and methods of proof, combinatorics, graphs and digraphs.

Objectives for the Course

By the end of the semester, a successful student will be able to do all of the following things:

- Write mathematical proofs at an introductory level
  - Frame your arguments in a clear, logical, and convincing fashion
  - Employ standard methods such as proof by induction and proof by contradiction
- Develop linguistic precision
  - Employing technical concepts in correct mathematical language, with correct syntax
  - Translate back and forth between mathematical/logical language and everyday natural language
- Understand and employ basic mathematical concepts and tools
  - Propositional and quantificational logic
  - Set theory
  - Relations and functions
  - Graph theory

Course Materials

1. **Textbook [SCH]:** *Mathematics: A Discrete Introduction*, 3rd ed., by Edward R. Scheinerman, Brooks/Cole/Cengage (2013).
   ISBN: 9780840049421
2. **Your own notes:** This course does not use slides. Regular note-taking is expected. Lectures will be recorded and posted to Canvas, but we can't promise anything with respect to recording quality on any given day.

Prerequisites and Expected Competencies

CS 11 or Math 32 is useful, but the primary outcome we are looking for is mathematical maturity, so that you are prepared to understand and write proofs, consider new number systems such as modular arithmetic, and be comfortable reading symbols such as set notation. You will need a few algebra skills here and there. This is not a programming course.

Requirements & Grading

Grades will be based on the following:

1. 30% Homework (10 assignments; all scores counted)
2. 40% Midterm examinations (3 exams; approx. 13.3% each)
3. 25% Final examination
4. 5% Participation/attendance during lecture sessions

Letter Grades: We use the following breakdown of letter grades and percentages:

98–100% A+	87–89% B+	77–79% C+	67–69% D+
93–97% A	83–86% B	73–76% C	63–66% D
90–92% A–	80–82% B–	70–72% C–	60–62% D–

Homework

Homework will be assigned regularly in the course. The homework due dates are listed on the schedule page. In general, students will have about one week for an assignment. Homework will be submitted using the Gradescope system, information about which will be provided when relevant.

Homework must be legible. If it is not type-written, then it must be so neatly written as to be equivalent. If illegibility makes grading slow and difficult, we will not grade the submission.

Students with extreme special circumstances must meet with the professor to make other arrangements to the scheduled homework and exams.

Students are welcome and encouraged to work together and discuss homework verbally. Every homework assignment should be written up separately and individually (please review the section on academic misconduct below). **Do not search online for solutions, and do not use AI tools to generate your work for you**.

Regrade requests for all homework assignments must be submitted within a week of the grades being released.

Exams

The exams will be in written format. Unless otherwise announced, exams will take place in the same room as the class itself. Example exams, to show the format and type of question, will be distributed before each exam occurs.

Students with extreme special circumstances must meet with the professor to make other arrangements to the scheduled homework and exams.

Students must submit an attempt at all exams in order to pass the course.

Regrade requests for all exams must be submitted within a week of the grades being released.

Policy on Late and Missing Work

For late homework assignments, handed in **within 24 hours after** the time at which it was due, a reduction of 10% will occur; if handed in **within 48 hours** of the expected time, a reduction of 20% will occur; **within 72 hours** the reduction will be 40%. No credit is given for assignments submitted after that point, without a documented reason. There is **no late deadline for exams**: they must be completed during the assigned exam period given.

Class Participation

Occasionally, we will distribute and collect lecture response forms during class. These will help us keep track of where everyone is with respect to the material, and will count for the participation portion of the grade. Up to 2 can be missed without penalty, but missing more than that will result in a reduction to the participation grade.

Policy on Collaboration

I encourage you to work together on the material. This is a great way to learn, and to share ideas. However, in order to actually learn something, it is important that you complete the real work on your own, unless specifically directed otherwise. It is perfectly fine for you to discuss the general approach to a problem with one another, work out how to understand a proof or concept, and to help one another with things like setting up and using LaTeX (if you decide to use it). However, it is **not okay** to copy proofs and other materials from anyone inside or outside of the class. While you can of course use online references to explain key concepts, and to learn what you need to succeed, don't simply copy answers you find online, and you should cite any such materials you consulted. Citation should extend to any AI tools that you endeavor to use to assist with homework, along with a clear explanation of how you used the tool, and how much of the work submitted should be credited to your own work, versus the work of the tool.

TL;DR: Do your own work. This is the only way to actually learn the material.

Piazza & Collaboration

When using the Piazza forum, the same sorts of considerations about collaboration are in play when posting questions and providing answers.

Questions may be posted as either **private** (viewable only by yourself and course staff) or **public** (additionally viewable by all students for the course registered on Piazza). Some issues warrant public questions and responses, such as: misconceptions or clarifications about the instructions, conceptual questions, errors in documentation, etc. Some issues are better with private posts, including: debugging questions that include extensive amounts of code, questions that reveal a portion of your solution, etc.

In addition, if you wish, remember that you can make your public posts **anonymous**, meaning that the instructors will know who is posting, but other members of the class will not. If that is something you want to do, for whatever reason, that is fine.

Please use your best judgment when selecting private vs. public. If in doubt, make it private.

Academic Misconduct

Students should read and know the Tufts policies on academic integrity. If a student does not understand these terms or any of the material listed on this page, it is his/her responsibility to talk to the professor. To be brief: do your own work, cite any sources from which you take ideas, and give credit where credit is due.

Inclusivity

In the classroom, we do not only expect participation, but also that everyone be treated with dignity and respect. Our perspective will be that everyone comes from a different background, each with their own personal history, identity, and background knowledge. Our knowledge will always be used to better everyone in the class. As an instructor, I have my own specific background and perspective on life, along with my own history of mental and physical challenges. I don't presume that my experience is the same as anyone else's, but will always do my best to meet all of my students where they live, to the best that I can. I'll probably make some mistakes, but I will be trying.

Policy on Sharing

This course is designed for everyone to feel comfortable participating in discussion, asking questions, learning, and facilitating the learning of others. In order for that atmosphere to be maintained, the recordings of our conversations will only be shared with the enrolled students in the class (not posted publicly), and it is prohibited for any of us who have access to the video to share it outside the course. Additionally, some readings are provided on a fair-use basis while taking this course, and are not to be distributed otherwise. This especially includes any posting or sharing of readings, videos, or other recordings on publicly accessible websites or forums. Any such sharing or posting could violate copyright law or law that protects the privacy of student educational records.

Writing Support

The StAAR Center for accessibility and academic resources offers friendly, experienced, non-judgmental writing support to writers at all levels of expertise through any stage in the writing process (for free!), and I highly recommend that you take advantage of this excellent opportunity.

Accessibility

Tufts University values the diversity of our students, staff, and faculty, recognizing the important contribution each student makes to our unique community. We are committed to providing equal access and support to all students through the provision of reasonable accommodations so that each student may access their curricula and achieve their personal and academic potential. If you have a disability that requires reasonable accommodations you can contact the StAAR Center (617-627-4539) to make an appointment to determine appropriate accommodations. Please be aware that accommodations cannot be enacted retroactively, making timeliness a critical aspect for their provision.

Counseling & Wellness

As a student, there may be times when personal stressors or emotional difficulties interfere with your academic performance or well-being. Counseling and Mental Health Services (CMHS) provides confidential consultation, brief counseling, and urgent care at no cost for all Tufts undergraduates as well as for graduate students who have paid the student health fee. To make an appointment, call 617-627-3360. Please visit the CMHS website to learn more about their services and resources.

CS 115 Database Systems, fall 2022

Syllabus v1.2.1

Course Description

Fundamental concepts of database management systems. Topics include: data models (relational, object-oriented, and others); the SQL query language; implementation techniques of database management systems (storage and index structures, concurrency control, recovery, and query processing); management of unstructured and semi-structured data; and scientific data collections.

Learning Outcomes

Upon completion of the course, students will be able to:

1. Explain database design concepts (including ACID, CAP, eventual consistency, and vertical/horizontal scaling) for both relational

2. Write queries that correctly and efficiently retrieve (SELECT), INSERT, UPDATE, and DELETE data in relational (SQL) databases, as well as queries that perform equivalent operations in several types of NoSQL databases.
3. Design a relational (SQL) database schema from a problem description and express the design as an Entity-Relationship (ER) or UML diagram or SQL language schema.
4. Enforce constraints on the data in relational (SQL) databases using schema constraints in the SQL language as well as using triggers/rules.
5. Prove that a relational (SQL) database is or is not in a particular normal form (3NF, BCNF/3.5NF, or 4NF), and transform a database schema into a particular normal form.
6. Choose the appropriate database system to use for a particular application, from relation (SQL) or several types of NoSQL databases.
7. Design databases for each of several types of NoSQL databases.
8. Use software design patterns that facilitate application code interfacing with database systems.

Prerequisites

- **CS 15 (Data Structures)** or *Graduate Student* standing
    - Grad students are expected to have had some sort of Data Structures course (e.g., CS 15, CS 205, or a Data Structures programming course from another college).
- Ability to program in some high-level language (e.g., C, C++, Java, etc.) and a willingness to learn new languages
    - You will receive explicit instruction on the SQL language in this

JavaScript (and possibly other languages), and you will be expected to get up to speed on your own with what you need to know to follow along with the examples. However, you will not be expected to write complete programs yourself in any of these languages; just a few lines at the most.

- In order to best understand some of the example and database concepts, you should be familiar with basic object-oriented programming concepts (classes, methods, abstraction/interfaces, etc.) as well as common data structures, such as trees

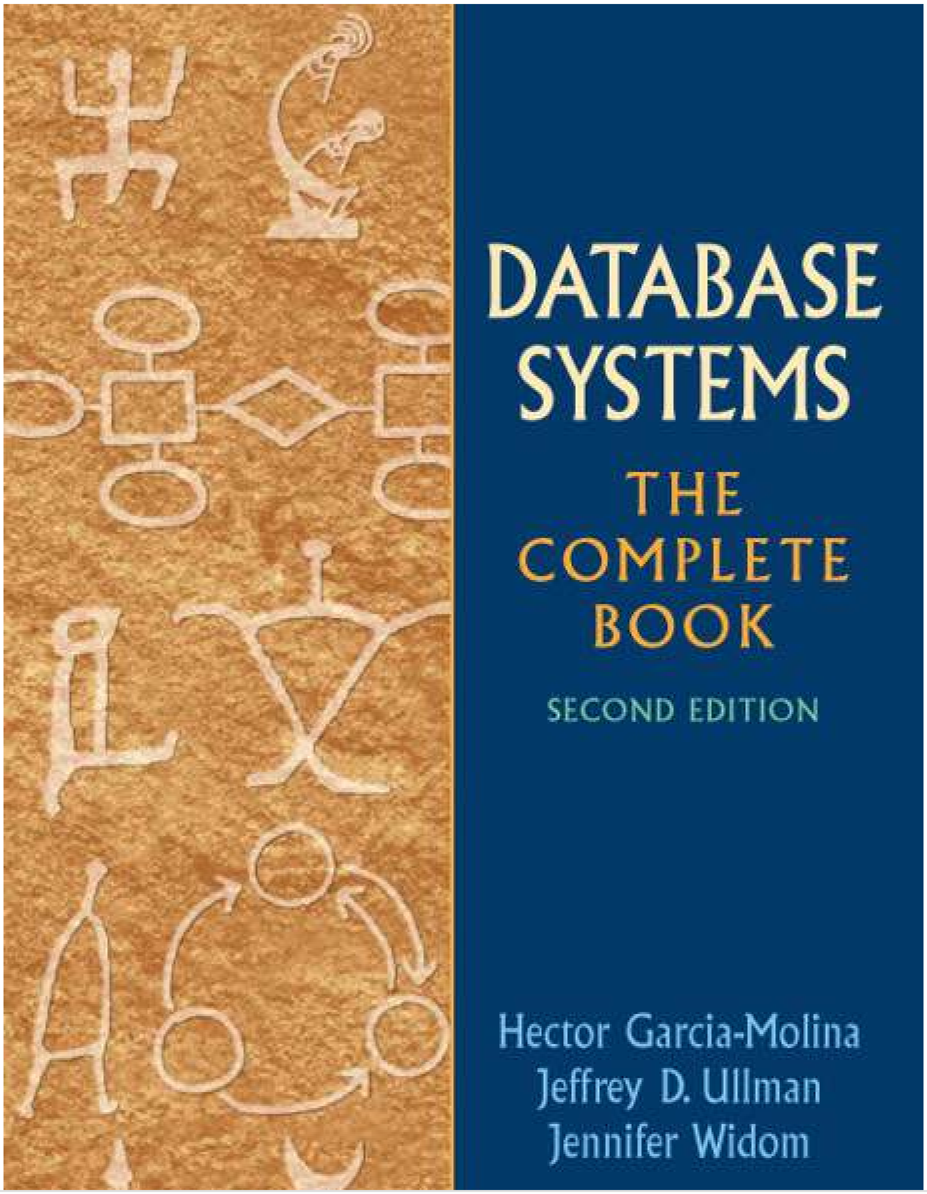
Course Materials

Required Textbook:

***Database Systems: The Complete Book, 2nd Edition*. H. Garcia-Molina, J. Ullman, and J. Widom (c) 2009. Pearson.**

ISBN-13: 9780131873254

(Hardback, paperback, and digital editions are all fine. New, used



rented/borrowed are also fine. But it must be the 2nd edition.)

You will also need a laptop or desktop computer at home that you can either install database software on, or run a virtual machine that has the database software on it. Windows, Mac, and Linux systems are all acceptable, and all required software is freely available online.

Class Times & Location

This class will be held *in-person* for fall 2021. (Keep an eye on your email for announcements form the university or AS&E schools about COVID-19 precautions, and for announcements from the course instructor.)

- Tuesdays & Thursdays, 6:00 – 7:15 PM
- Joyce Cummings Center, room 160

The first class will be Tuesday, Sept. 6, 2022. The last day of class will be Thursday, Dec. 8, 2022. The final exam will be **Friday, Dec. 16, 2022, at 7:00 pm**.

Instructor & TAs

Dave Lillethun is the instructor. TAs are Tomislav Zabcic-Matic and Mona Ma.

Office hours and contact information will be posted in Canvas.

Assessments

Take-Home Tests

There will be several tests (every other week, 6 in total), which you will tests. They will not be proctored and will not be strictly timed, but will be released at a certain time and due at a certain later time. You may work on the test any time before the deadline and take as long as you like, up to the deadline. Materials you are allowed to use when working on tests are:

- While taking the test, you may refer to the textbook, your notes (including previous tests, quizzes, class exercises, etc.), the Internet, and generally any other pre-written/recorded sources.
- You *may **not** collaborate or discuss* test questions or answers with anyone else (whether they are a student in this class or not). This also means that you may not post online (e.g., Stack Overflow, Reddit, etc.) to ask questions pertaining to the test; you may only use information online that already existed before you looked at the test.
- **You must write all answers in your own words.** Even though you may refer to other materials to help you figure out the answers, you may not plagiarize those materials on your answers. Even making minor modifications after copying is forbidden. You must *completely rewrite/rephrase it **using your own words***.
- If ever in doubt, ask the instructor *before* you do something. It is better to ask for permission than forgiveness, in this case.

In-Class Quizzes

There will be several quizzes (every 2-3 weeks, 5 in total), which you will take in class. You will be given **30 minutes** to complete the quiz at the beginning of class on days when quizzes are given, and normal class will be held for the rest of the period following the test. **Quizzes are *cumulative***, and you may be asked about any material covered in any lecture or required reading assignment from the textbook

Quizzes will be proctored, and you *may **not** use any calculators, other electronic devices, books, or notes* ***except*** for a **one-page note sheet**. You will be allowed to prepare a single letter size (8.5″x11″) sheet of paper with notes that you may use during the exam. You may write on both sides, and there are no rules about text size or margins, but it *must be **hand written*** (unless a disability accommodation allows otherwise). However, you will not be allowed to use any magnification devices (except for routine corrective eyewear, such as glasses), so you must write it such that you will be able to read it. While you are welcome to discuss what to put on the note sheet with other students, **each student must hand write their own copy of the note sheet**. *Photocopies, and other kinds of copies, will **not** be permitted*.

You will be required to turn in your note sheet with your quiz. Among other reasons, the will force you to rewrite a new one for each quiz, which is beneficial for several reasons:

- **The act of writing things down will help you to remember.** Therefore, rewriting the things you need help remembering several times will help you remember them better than just writing them once. (Also, hand writing it will help you remember better than typing would.)
- There will be **new material on each quiz** that you'll want to add to your notes sheet.
- You should also remove old material from the notes sheet. If you find that you were able to remember something without the notes sheet, you may decide to remove it. More importantly, **you'll want to make room for the new material** that you're adding.

Final Exam

The final exam will take place at **7:00 pm on Friday, Dec. 16** in the usual classroom. It will have the same rules, format, and types of questions as the quizzes (including the notes sheet), with the same types and formats of questions, except that it will be fully cumulative. All material from the entire class will be eligible for exam questions. The length of the exam will be approximately twice that of a quiz. (So in essence, it's a longer, more fully cumulative quiz.)

Late Work Policy

In this class, each students starts the semester with **5 "late tokens"** that are each worth an extra day (24 hours) on a take home test assignment. (With the exception that you will get Saturday and Sunday as a bundle for 1 token. Since tests will be due on Fridays, that means 1 token gives you until Sunday, 2 tokens until Monday, and an additional token for every day past then that you need.) There are no penalties for lateness as long as you use a sufficient number of late tokens. If you will not have work completed by the deadline, then **email the instructor** (contact information is in Canvas) ***before* the deadline** and tell him how many extra days you need. A corresponding number of late tokens will be deducted. You don't need to provide any excuses, just the new date when you will submit the work. As long as you send that email before the assignment deadline, your late submission will be graded for full credit. (However, if there is an emergency situation that makes it impossible or unreasonable to contact us before the deadline, then just send the email as soon after the deadline as is reasonable.)

Note that the quizzes and final exam are scheduled to occur in specific class periods, and therefore the deadline cannot be extended. However, make up quizzes and exams can be provided if you miss it due to a

miss any of these.

Course Grades

Each test, quiz, or exam will *not* be given a percentage score or letter grade, but rather each individual question will be rated as "Exceeds Expectations", "Meets Expectations", "Needs Improvement", or "Not Assessable" ("N/A" for short). These mean the following:

- **Exceeds Expectations** – This assessment shows that you have significant proficiency in the relevant topic, beyond the minimum expected by the course learning expectations.
- **Meets Expectations** – This assessment shows that you have some proficiency in the relevant topic, in line with the course learning expectations.
- **Needs Improvement** – This assessment shows some knowledge but does not yet demonstrate full proficiency in the topic.
- **Not Assessable** – This assessment is too incomplete to accurately assess whether you have met expectations or not, or it shows little evidence of knowledge or skills. This could be, for example, a question in which not all parts were attempted fully, was left blank, or was partially answered but too much was missing to really be able to assess the student's knowledge/skills.

Here is what you need to do to earn each letter grade. You must meet *all* of the criteria for a letter grade in order to get that grade.

- A
    - Meets Expectations (or higher) on a total of **at least 90.0%** of test questions, and
    - Exceeds Expectations on **at least 30.0%** of test questions, and

  - Exceeds Expectations on **at least 30.0%** of quiz & exam questions
- B
  - Meets Expectations (or higher) on a total of **at least 70.0%** of test questions, and
  - Exceeds Expectations on **at least 10.0%** of test questions, and
  - Meets Expectations (or higher) on a total of **at least 70.0%** of quiz & exam questions, and
  - Exceeds Expectations on **at least 10.0%** of quiz & exam questions
- C
  - Meets Expectations (or higher) on a total of **at least 55.0%** of test questions, and
  - Meets Expectations (or higher) on a total of **at least 55.0%** of quiz & exam questions
- D
  - Meets Expectations (or higher) on a total of **at least 50.0%** of test questions, and
  - Meets Expectations (or higher) on a total of **at least 50.0%** of quiz & exam questions

Students who meet all the criteria for a letter grade, **plus at least 10.0%** (percentage points) more than required Exceed Expectations on *each* of the test questions category and the quiz & exam questions category, **will receive a plus (+)** on their grade. (e.g., B+ for 30% of test questions and 40% of quiz & exam question, or C+ for 10% of test questions and 10% of quiz & exam questions Exceed Expectations.)

Students who are **within 5.0%** (percentage points) of the required number of questions Meeting or Exceeding Expectations, on *each* of the test

Expectations on *each* of the test questions category and the quiz & exam questions category, **will receive that letter with a minus (-)**, rather than the next lower letter. (e.g., Exceeding Expectations on 10% of test questions and 20% of quiz questions, while Meeting (or Exceeding) Expectations on 80% of test questions and 80% of quiz & exam questions is a B-. However, if you instead Exceed Expectations on only 10% of quiz questions, then that's a C+, and if you Meet (or Exceed) Expectations on only 79% of test questions, then that's a C no matter how many questions you Exceeded Expectations on.)

Students who do not meet the criteria for at least a D- will receive an F.

Course Policies

Students in this class are responsible for reading, understanding, and following all of the course policies listed here. For fall 2021, there are also additional COVID-19 policies for the course. By continuing to take this course, you indicate your agreement to follow all the policies. If there is any policy that you do not understand, please ask. Ignorance will not be accepted as an excuse for violating any policies.

Changes

This syllabus and any policies for this course are subject to change during the semester in response to changing conditions. Such changes are at the sole discretion of the course instructor. If any changes are made, the appropriate documents will be updated and the change will be announced to the entire class in a timely fashion.

This document has a version number at the beginning to help you tell when example, so you don't need to re-read the entire document just to find a small change.)

Change Log

v1.0: Initial version.

v1.1: Updated course grade calculations after mid-semester evaluation

v1.2: Updated course grade calculations for the final curve (end of semester)

v1.2.1: Added ".0" to percentage values in course grade information, to clarify that these are exact decimal values (not integers) so there will not be any rounding.

Oliefabriksvej 29, 43, 2770
Kastrup, Denmark

info@example.com
+1234567890

Facebook / Instagram
LinkedIn / Google

Designed with WordPress

CS 116: Introduction to Security

Tufts University Department of Computer Science, Spring 2025

Course Description

A holistic and broad perspective on cyber security. Attacking and defending networks, cryptography, vulnerabilities, reverse engineering, web security, static and dynamic analysis, malware, forensics. Principles illustrated through hands-on labs and projects, including Capture The Flag (CTF) games.

Sections

1. In-person, undergrads and grads: Tuesdays, 4:30 - 5:45 PM EST in Cummings Center Room 270; Thursdays on Twitch, 4:30 - 5:45 PM EST
2. Online Master's in Computer Science: live sessions on Wednesdays, 5:30 - 7:00 PM EST (via Zoom)

Instructor

- Ming Chow, ming.chow@tufts.edu
- Office Hours: Tuesdays, 3:00 - 4:15 PM EST in JCC, fourth floor --by the kitchen area

Prerequisites

- CS 15 or Data Structures equivalent course. Recommended (not required) that you have taken CS 30 or 40. Please disregard prerequisite that CS 40 is required as listed in the University's bulletin as they are incorrect!

Hardware and Software for This Class (on your personal computer)

Absolute Requirements

- A modern web browser (e.g., Firefox, Google Chrome, Chromium, Safari, Microsoft Edge)

- A command line interface to run Unix/Linux commands (e.g., macOS, Windows with Linux Subsystem, a Linux-based virtual machine, a Docker container)

List of Security Tools That Will Be Used in Course

The following is a list of security tools that will be used in the course. All of these tools are platform-independent.

- Wireshark
- Nmap
- Netcat
- Python
- Scapy
- John the Ripper
- Burp Suite
- apktool and Java as apktool requires Java

Assessment

- Labs (80%)
- Quizzes (20%; there will be two)

Course Infrastructure

- Lab submissions: Canvas (Tufts UTLN required)
- Quizzes: Canvas (Tufts UTLN required)
- Announcements: Piazza
- Discussions: Piazza

Syllabus

Important note: always follow Canvas for due date on labs!

Topic 1, starting week of January 15th	• Course Introduction - By the end of this week, students will learn many of the fundamental Linux commands, an important skill for any good security practitioner, by playing	• Lab 1: Working with the Command Line ○ Publicly accessible version of Lab 1

	Capture The Flags via OverTheWire. Students will remember the three principles of the CIA triad, critical to any organization's security infrastructure. • Readings and Videos • Video on YouTube with Closed Captioning: The Command Line Interface	
Topic 2, starting Tuesday, January 21st	• Networking and Packets - By the end of this week, students will be able to dissect packet captures (PCAPs), network traffic. • Readings and Videos • Thursday, January 23rd: Packet Analysis Using Wireshark ○ On Twitch ○ On YouTube with Closed Captioning	• Lab 2: Packet Sleuth ○ Publicly accessible version of Lab 2
Topic 3, starting Tuesday, January 28th	• Attacking Networks - By the end of this week, students will perform network reconnaissance and port scanning, and build a rudimentary Security Information and Event Management (SIEM) / intrusion detection system (IDS). • Readings and Videos	• Lab 3: Scanning and Reconnaissance ○ Publicly accessible version of Lab 3 • Lab 4: Snake Oil, The Incident Alarm ○ Publicly accessible version of Lab 4

	• Thursday, January 30th: Reconnaissance using Ping, Netcat, and Nmap ◦ On Twitch ◦ On YouTube with Closed Captioning • Thursday, February 6th: Distributed Denial of Service (DDoS) Attacks and Scapy feat John Hammond ◦ On Twitch ◦ On YouTube with Closed Captioning	
Topic 4, starting Tuesday, February 11th	• Cryptography - By the end of this week, students will be able to crack passwords on a Linux or Windows system, use one-way hash functions, and briefly describe how Transport Layer Security works. • Readings and Videos • Thursday, February 13th: Passwords and Password Cracking with John the Ripper ◦ On Twitch ◦ On YouTube with Closed Captioning	• Lab 5: Password Cracking Contest opens, due on Friday, March 7th at 11:59 PM PST
Topic 5, starting Tuesday, February 18th	• Vulnerabilities - By the end of this week, students will know the difference between CVE and CWE. • Readings and Videos	• Quiz 1 open on Monday, February 17th; due Sunday, February 23rd at 11:59 PDT

Topic 6, starting Tuesday, February 25th	• Web Security - By the end of this week, students will able to perform and defend against the following attacks: Cross-Site Scripting (XSS), SQL injection, Cross-Site Request Forgery (CSRF), session hijacking, cookie tampering, directory traversal, command injection, remote and local file inclusion. • Readings and Videos • Thursday, February 27th: SQL Injection and Web Proxies ◦ On Twitch ◦ On YouTube with Closed Captioning • Thursday, March 6th: Vulnerability Scanning, Exploitation, Badness-O-Meter ◦ On Twitch ◦ On YouTube with Closed Captioning • Thursday, March 13th: The Mistakes You Can't Make ◦ On Twitch ◦ On YouTube with Closed Captioning	• Lab 6: The XSS Game ◦ Publicly accessible version of Lab 6 • Lab 7: Gain Access to Website ◦ Publicly accessible version of Lab 7
Topic 7, starting Monday, March 24th	• The Capture The Flags (CTF) Game Played Online - By the end of this week,	• Lab 8: The CTF Write Up ◦ Publicly accessible

	students will be able to find and take advantage of a number of vulnerabilities on a live web application. - Readings and Videos	version of Lab 8
Topic 8, starting Tuesday, April 1st	- Static and Dynamic Analysis - By the end of this week, students will be able to perform static analysis and dynamic analysis scans on software, write a technical risk analysis that is communicated to upper management. - Readings and Videos - Thursday, April 3rd: Really, Really Bad Code and Static Analysis ○ On Twitch ○ On YouTube with Closed Captioning	- Lab 9: Technical Risk Analysis
Topic 9, starting Tuesday, April 8th	- Malware - By the end of this week, students will be able to describe types of malware, see certain malware behaviors, scan and analyze malware, reverse engineer Android apps to determine if they are malicious. - Readings and Videos - Thursday, April 10th: Malware and Malware Analysis ○ On Twitch	- Lab 10: Android Malware Analysis ○ Publicly accessible version of Lab 10

	◦ On YouTube with Closed Captioning	
Topic 10, starts Tuesday, April 15th	• Forensics and Incident Handling - By the end of this week, students will be able to acquire data from a disk (e.g., USB drive) using **dd**, analyze image of disk from `dd` using forensics tools, and recover deleted files off a disk. • Readings and Videos • Thursday, April 17th: Forensics ◦ On Twitch ◦ On YouTube with Closed Captioning	• Quiz 2 open on Monday, April 14th; due Tuesday, April 22nd at 11:59 PM PDT
Topic 11, starts Tuesday, April 22nd	• ~~The Future: Nihilism or Hope? - By the end of this week, students shall debate and ponder the hard questions in security, and be able to argue multiple viewpoints.~~ Lessons Not Learned: We Can't Even Get the Basics Right • Readings and Videos • Thursday, April 24th: Opportunities, Where Do You Go From Here ◦ On Twitch ◦ On YouTube with Closed Captioning	

Topics That Will Not Be Covered In This Course

- Social Engineering
- Privacy

Frequently Asked Questions

Q: Is there a textbook for this course?

A: No

Q: Are there teaching assistants (TAs) for this course?

A: No

Q: What is the workload of this course?

A: Here is a list of all the labs with expected length and difficulty:

- Lab 1: Working with the Command Line, Short (1 hour max) to Long (3+ hours) --you can put in as much time as you want on this lab
- Lab 2: Packet Sleuth, Medium (1 - 3 hours)
- Lab 3: Scanning and Reconnaissance, ~~Very short (30 minutes)~~ Your mileage may vary on this lab, could be 30 minutes, could be 2 hours or more. **NOTE: This lab cannot be made publicly available because an actual target is used.**
- Lab 4: Python and the Incident Alarm, Long (over 3 hours) to Impossible
- Lab 5: The Password Cracking Contest, If you crack all the password hashes (read: good luck with that), you will receive an automatic "A" in the course
- Lab 6: The XSS Game, Medium
- Lab 7: Gain Access to Website, Very short. **NOTE: This lab cannot be made publicly available because an actual target is used.**
- Lab 8: The CTF Game, One week --team based. **NOTE: This lab cannot be made publicly available because an actual target is used.**
- Lab 9: Technical Risk Analysis, Short to Medium.
- Lab 10: Android Malware Analysis, Short to Medium

Q: Does this course count towards the M.S. in Cybersecurity and Public Policy?

A: Yes

Q: Does this course count towards the M.S. in Software Systems Development?

A: Yes. In fact, this is one of the four required courses for the M.S.

Q: Did you remove information on using Kali virtual machine for this class? If so, why?

A: Yes. After all these years, it was more trouble than it was worth. Further reasons:

1. Accessibility. For students who are visually impaired, using a virtual machine can be very difficult.
2. Not all students have a capable laptop. Sometimes due to financial reasons, some students use Chromebooks. The tools required for this course can be installed natively on macOS, Windows, and Linux.
3. Performance. Sometimes, using a VM can be very slow. A VM also do not use native drivers (e.g., for networking).
4. Hard disk space requirement: at least 10 GB necessary.
5. Apple M1 Macs cannot run most Intel x86 virtual machines.

Q: Is Piazza used in this course?

Yes, quite a lot

Q: Why is there a course website and a course Canvas? If you say "it is a nuisance for students to use multiple websites and services for one course", what gives?

This course website serves a few critical purposes. Years ago, I made a decision to make all the readings, slide decks, and most of the labs publicly available. The reasons: (1) to show that Tufts is serious and is working on Cyber Security matters, (2) to provide learning material to the public on Cyber Security as the Cyber Security education problem is very dire, (3) for recordkeeping on what is taught and not taught in this Security class --this comes up often when we speak to industry and organizations who want to work with Tufts on Cyber Security-related matters. The Canvas site for this course isn't made publicly available. Even if Canvas site was made publicly available, content is behind a walled garden, and (4) for redundancy if Canvas goes down.

Q: I have not taken a course on Networks (CS 112), Operating Systems (CS 111), or Computer Architecture (CS 40) yet. Is that a problem?

No. Cyber Security is a very broad field and it is impossible for anyone, even professionals, to know everything. What is important for you is to start thinking about Security.

Q: Will videos be recorded in case I miss class?

If you miss the in-person Tuesday classes due to illness or personal matters, you can always watch the recorded Wednesday sessions for online Master's students (Wednesday sessions are always recorded). Tuesday and Wednesday sessions are practically identical. Thursdays are on Twitch which are recorded and also exported to YouTube.

Q: If I am taking this course for professional purpose, can I have a tuition reimbursement letter or certificate?

A: Absolutely! It's a nice tuition reimbursement letter, hand signed!

If you have read this far, send me an email (ming.chow AT tufts DOT edu) with the subject "¯_(ツ)_/¯" to earn a reward.

Course Policies

Labs

- With the exception of the password cracking lab and CTF writeup, all labs for a given topic, are due on a Sunday at 11:59 PM PDT (that is, Pacific Time).
- With the exception of password cracking lab, the CTF game, and quizzes, you are granted an automatic extension of 24 hours at no cost (i.e., grace period). A lab or quiz submitted after the grace period will not be accepted.
- No extension tokens.

Accessibility Statement

Tufts is committed to providing equal access and support to all qualified students through the provision of reasonable accommodations so that each student may fully participate in the Tufts experience. If you have a disability that requires accommodations, please contact the StAAR Center (formerly SAS) staarcenter@tufts.edu or 617-627-4539 to make an appointment to determine appropriate accommodations. Please be aware that accommodations cannot be enacted retroactively, making timeliness a critical aspect for their provision. You can learn more about the StAAR Center at https://students.tufts.edu/student-accessibility-services.

Expectations and Structure of This Course

This course will be a fun one for sure. A few notes on the expectations and structure of this course:

1. You are responsible for your own learning.

A very important point: if you want everything gone over in lecture or in notes, then this is not the course for you. More importantly, that's not how things work in real life.

2. You will learn by doing.

Each week, there will be at most three labs to hone your skills and to aim at the crux of the matter for the week. Here's an analogy: you don't learn how to cook simply by just reading cookbooks and watching YouTube videos. You learn by making, using your hands, and making mistakes.

3. You will learn by asking questions.

It is your responsibility to ask questions early and to ask for help...

4. ...and I expect discussions online to be very active and civil.

Share thoughts and respond to other people's questions. I will be online constantly. It is no secret that I respond very quickly unless I need to be away.

created by human
not by AI

CS 160-01: Introduction to Algorithms (Spring 2025)

Go to CS 160 schedule

General Information

- **Instructor:** Karen Edwards
- **Lectures:** MW 9:00-10:15 (R+ block) in Barnum LL08
- **Attendance:** There will be a lecture feedback sheet approximately once a week in lecture to measure understanding and participation.
- **Office Hours:** See Piazza.
- **Exams:** Dates can also be found on the schedule page.
  - Exam 1 and Exam 2: 75 minutes during regular class. Dates: Wed Feb 19 and Wed Mar 26.
  - Final Exam: CONFIRMED FOR Wed, May 7, from 7-9 PM, location TBD
- **Homework:** The homework, typically covering the Tuesday and Thursday lecture and the Thursday-Friday recitation is due by 11:59pm on the following Tuesday.
- **Canvas:** Canvas is being used **only** to share video recordings of lectures. There are no guarantees about these videos. If you have to miss a class, a better approach is to rely upon friends with good note-taking skills.

Quick Links:

Prerequisites
Course Logistics: Syllabus, Textbook, Tools
Textbook, other materials
Instructional Team and Office Hours
Expected Work: Participation, Quizzes, Homeworks, Project, Exams
Grading Policy and Clarifications
Tips on doing well
Important Tufts Policies and Resources
Have Questions?

Prerequisites

The prerequisites for this class are CS 15 and (CS/MATH 61 or MATH 65). If you have not successfully completed both courses either at Tufts or elsewhere and if you are enrolled in CS 160, please contact your instructor. We will be glad to help with references for you to study, but please understand that learning the prerequisite topics is mainly your responsibility.

This class also assumes familiarity with a few math topics such as logarithms, manipulation of summation notation, and summation of various geometric series. Be prepared to brush up on such topics as needed.

Course Logistics

Syllabus

- This page (and the schedule page) function as the course's syllabus.

Textbook

- The course is heavily based on the following book:
  [CLRS] T. H. Cormen, C. L. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (4th edition).* MIT Press, 2022.
  - We will also provide section numbers from the 3rd edition, if the topic is in there. (If there is something covered in the new edition that is not there in the older one, or is presented in a different way, be aware that we will be using that newest edition, and your experience is likely to be better if you can gain access to that one.)
  - More info at MIT press. Many students find they can learn more easily when they have access to a hard copy (even if it is an older edition).
  - You may also use this link to access a digital copy of the 3rd edition from Tufts.
  - Two copies of the 3rd edition have been permanently reserved on reserve in the Tisch library for students to access.
  - **Note:** The book is a comprehensive introduction to algorithms and contains many wonderful topics beyond what will be covered in this course.

- For further reading, any of these additional texts is recommended as an extra reference (all of them should also be available in the Tisch library, please alert us if not).
  Several students in the past have found the Skiena text particularly readable and have enjoyed his particular perspective, and yet each of these supplemental texts has had fans!
  - [KT] J. Kleinberg, E. Tardos. *Algorithm Design,* Addison-Wesley, 2005.
  - [DPV] S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani. *Algorithms,* McGraw-Hill, 2006.
  - [S] S. Skiena, *The Algorithm Design Manual,* (2nd edition) Springer, 2010.
    - Use this link for an online version from Tufts of the second edition. See here for errata.
  - [BVG] S. Baase and A. Van Gelder, *Computer Algorithms*, 3rd edition, Addison-Wesley, Reading, MA, 2000.

Course Tools

- We will use Piazza for class discussions, announcements, polls, handouts, assignments, and so on.
- Homework will be submitted and graded using Gradescope.
- Video recordings of lectures (if successful, no guarantees) will be available on Canvas.

Instructional Team:

Lead Instructor:

- Karen Edwards

Grad TAs:

- Jacob Boerma
- Jocelyn Garcia

Teaching Fellows:

- Dan Patterson
- Kimaya Wijeratna

CAs:

- Daniel Peng
- Eli Morton
- Johnny Tan
- Kevin Yu
- Lakshita Jain
- Marlon Fagundes Pereira Junior
- Megan Yi
- Meghan Kelly
- Merwan Malakapalli
- Peter Scully
- Rafeed Anwar
- Saška Barancikova
- Shayne Sidman
- Stephen Burchfield

Office Hours:

- Office hours are a great place to hang out and get work done, whether or not you have any questions yet. We have a large pool of motivated and talented TAs. Please don't be afraid to join them in office hours as needed to help you further understand the material.
- List of office hours is available on Piazza. Any cancellations, one-time location changes or additional hours will also be noted there.

Expected Work

Participation: Classes, Recitations, Quizzes/Feedback Forms:

- Students are expected to attend class and complete the related reading assignments.
- Students are expected to attend and participate actively in recitation sessions (attendance will be taken; we will drop two). These sessions provide an invaluable opportunity to practice the material and solidify learning.

- There will be lecture feedback sheets approximately once a week in lecture to measure attendance and understanding. We will drop two. Part of your grade will be based on this participation.

Homework:

- Most weeks, you will be given an assignment to practice the material presented in class and discussed in recitation. Due 11:59pm on Tuesday.
- Individual completion and regular timely submission of the homework with full attribution of sources is a prerequisite for passing this course.
- To obtain full credit, you must justify your answers. When describing an algorithm, do not forget to analyse its running time and justify why the algorithm is correct.

Collaboration and Integrity on Homework Submissions

Although you may discuss these problems in the preliminary stages with others, **submitted work should be done individually** and written in **your own words**. If you have any discussions with others, whether students, friends, TAs or faculty, relative to a homework problem or if you gain information from a written source (e.g., website) other than your own notes from lecture or the textbook for the course, you **must identify your collaborator(s)/source(s)** in writing on your homework submission. Failure to cite your sources constitutes an academic integrity violation and may be reported to the Dean.

How to write proofs:

On Piazza there is a short summary containing the key points on how to write proofs. In addition to our guide, there are various materials on the web to check out. We encourage you to look around for those resources whenever you have time in the beginning of the semester.

How to submit homework

- Homework assignments will be available on Piazza. Homeworks are turned in via Gradescope.
- Homeworks **must be typeset** using your preferred software -- LaTex is one option, and it is highly recommended. It is also required for the first homework. In general, you may handwrite or draw diagrams, trees, and graphs, which you must include/embed in the PDF. Any associated math, analysis, or descriptions of the diagrams should be typeset.

Late homework policy:

- You are allowed **~~four (4)~~ now six (6) "tokens"** to be used at your discretion. Each token accounts for a **24-hour automatic extension** on the homework.
- Up to **two (2) tokens** can be used on the same assignment (for a 48-hour maximum extension). After that, if you want to use more than 2 tokens for a longer extension, you need to ask either in Piazza (to "instructors") or in office hours.
- Tokens will be applied automatically based on the Gradescope timestamp of your last submission. You do not need to notify staff that you are using tokens.
- Make sure to keep track of how many tokens you have used. (You can see late submissions on Gradescope).
- Our calculations have a small grace period (of roughly 10 minutes) to account for last minute changes and/or differences in clocks. This grace period is automatically applied to all submissions.
- If you run out of tokens (or submit homework more than 48 hours late), late homework will be counted as 0. However ....
- At the end of the semester we will calculate how many tokens are used by each student. If a student missed the deadline once by a couple of minutes (even after taking into account the tokens and the

grace period), then we will be lenient. Any other case will count as 0, which will be applied to the lowest late homework(s) to minimize impact. So it is always worth turning in homeworks even if you are late and already used all your tokens.
    - In short, be punctual and you will save all of us from troubles! If you have a valid reason for not submitting homework or need to request a longer extension, you should notify your Dean (and, if applicable, Health Services). Decisions on any extension requests will be made in consultation with your Dean.

Exams:

- The midterm exams and the final exam will include questions similar to those included on homework assignments or recitation problems and will draw on material covered in lecture and/or in the reading.
- Exam dates and times are listed under General Information and on the schedule page.
- **There are no make-up exams, so check the exam schedule before making travel arrangements.**

Grading

Your final grade percentage will be the **maximum** of the following two formulas:

> **Overall grade = 25%(Homework) + 5%(Participation) + 2x20%(per midterm exam) + 30%(Final exam)**
>
> OR
>
> **Overall grade = 25%(Homework) + 5%(Participation) + 2x15%(per midterm exam) + 40%(Final exam)**

Clarifications:

- No homeworks will be dropped.
- Participation is half recitation attendance, half lecture attendance (as measured by turning in class gradescope quizzes.) Two recitations and two class gradescope quizzes will be dropped, i.e., you can miss up to 2 of each for any reason. We do not distinguish between "excused absences" and other absences; all such requests are covered by dropping 2 for everyone. If you have valid reasons for missing 3 or more recitations, or 3 or more class gradescope quizzes, please email the instructors.
- Exam grades will not be curved individually, but the final overall letter grades of the course will be adjusted as needed. In particular, anyone in this class whose overall average is at or above the median will earn at least a B+.

Regrade requests:

If you believe any part of your work on Gradescope was graded incorrectly, we want you to let us know via a regrade request on Gradescope. Regrade requests should be concise, directed, and respectful. You should point out a specific aspect of the assignment that was graded incorrectly, and explain why in your own words. We also reserve the right to deduct points if we notice additional errors during the regrading. The person who regrades your submission will send you a response. If you have further concerns thereafter, you may contact one of the TFs or Graduate TAs at office hours or through a private Piazza post. Regrades must be submitted within **one week** of release of the grades, unless stated otherwise.

Tips on getting the most out of this course:

- Come to every class prepared. At a minimum, you should review the previous lecture, and you may also benefit from looking over the reading beforehand. Simply showing up will not suffice for your learning.
- If the pace of the lecture feels too fast, consider reading some basic material beforehand, perhaps in one of the auxiliary texts.
- When studying, try to re-derive the solutions on your own, rather than verifying what is written. You would like to "own your knowledge".
- Please don't cram. This material needs time to sink in. It cannot be learned in a couple of days.
- Spend regular time on this **core course in computer science**. It will require as much time as other core courses such as CS40 and CS105 where you may spend significant time programming. In CS160, you should spend an equivalent amount of time developing a deep understanding and mastery of the core theoretical concepts and algorithm paradigms being presented.
- After doing the reading and attending class, brainstorm about homework problems, perhaps with a small study group including 2-3 other students, informally gathering ideas and without taking notes. Then remember to go off by yourself and work independently at writing up the solutions. Make sure to cite the other members of your study group.
- Use all of the course resources. Actively participate in recitation. Attend office hours when you have questions. If additional questions arise, post them to Piazza. Solve extra practice problems from the book (these may vary in difficulty). Make your own practice problems (e.g., draw an arbitrary graph and find shortest paths).
- Read all the grading feedback on your homework as well as any published solutions (e.g., recitation). The TAs are devoting many hours to grading and giving useful feedback on your work.
- Avoid memorization. Instead, focus on understanding and then being able to explain a concept or prove a theorem in your own words.
- One of the most helpful strategies is to imagine that you are teaching this material to your parent or to a cousin or to a next-door neighbor. If you don't have the details right and don't fully understand the solution yourself, you will confuse them. (Try and actually practice explaining the concepts!) This is another reason we encourage you to collaborate in small groups, to be active at recitations and to attend office hours, since these are good opportunities to discuss the course materials with other people.

Tufts University Policies

Faculty, students, and staff are jointly responsible for the policies below. Please read carefully and do not hesitate to contact us if you have further questions.

- Academic Misconduct Policy
- The Tufts Non-Discrimination Policy

Equal access

If you have a disability that requires reasonable accommodations, please contact the Office of Accessibility Services at The StAAR Center.
(Note: Academic tutoring is also offered by the StAAR Center for those who need tutoring through Tutor Finder)

Please be aware that accommodations cannot be enacted retroactively, making timeliness a critical aspect for their provision.

Academic Integrity

- It is **not** acceptable to copy solutions from **any** source for **any** work submitted in this course.
- If you cheat on an exam by consulting any source other than the allowed page of notes, the most likely penalty is a grade of F in the course.
- On a homework or project assignment, students may consult a variety of sources at the outset and groups of people may work together, discussing and strategizing how to solve the problems, **subject to the following conditions**: each person must write and submit their solutions in their **OWN** words; **every person and/or text and/or website** consulted in the process of completing the assignment must be **accurately cited** on the homework paper submitted.
- If, for whatever reason, your homework solutions end up matching solutions from any other source in a non-coincidental way, this constitutes cheating and will incur consequences.
- In case of doubt, we encourage you to reach out to the instructor.

Have Questions?

This website is a great source of information for this class. Before inquiring, please check for the answer here (e.g., when is the exam, how do tokens work, etc).
If you still have questions, we can be reached as follows:

- **In class:** Just come and chat with the instructor before/after the lectures.
- **Office hours:** We have a large pool of TAs to cover most work hours. Please keep in mind that the further away you are from a deadline, the less waiting time you will have.
- **Piazza:** The main advantage of posting your question on Piazza is that the TAs and/or your classmates will also see that question and can answer it much faster. Make sure to use the search button before posting because your question might have already been asked! If the question needs to be private, please post to "Instructors" (this will reach Karen and all the TAs).
- **PLEASE NOTE:** Your communication with TAs should be at office hours and over Piazza. **Please do NOT email TAs directly** -- their work for the department as TAs should be conducted at office hours and over Piazza, keeping their personal email addresses for their own use as students.
- **E-mail:** You can contact the course instructor using the syntax first.last@tufts.edu. However, other methods may be better than emailing in order to get the fastest response.

Course Aims and Description

This is a computer science elective aimed at upper level undergraduates and graduate students. Upon the completion of the course, students will be able to:

- Comprehend the biological background, nature, and relevance of computational problems in molecular biology.
- Assess the efficiency of computational methods for handling data-rich problems in the field.
- Understand computational techniques and probabilistic models for working effectively with large data sets.
- Discuss and evaluate tradeoffs involved in choosing how to tackle hard computational problems.
- Develop experience applying theoretical CS material in a practical setting.

Mastery of these aims will be achieved and assessed through readings, discussions, problem sets, algorithm implementation or data analysis assignments, two in-class midterms, and a final exam. About half of the course will focus on molecular sequences and sequence manipulation; the rest will focus on issues of interpretation, which require more complex data and methods. We will talk about scalability and how and when approximate solutions are appropriate. In addition, we will introduce some ongoing areas of research in the fields of bioinformatics and computational biology.

Students will be expected to contribute to class discussion and group activities, to do the assigned reading, and to read supplementary background materials as they find necessary.

Course Staff and office hours:
Professor Donna Slonim (she/her) is the course instructor.

CS PhD student Blessing Kolawole will be our graduate teaching assistant. CS Major Adhvith Reddy is our undergraduate course assistant. TA office hours: Mondays at 7-8:30pm and Thursdays at 12:30-2pm, in JCC 349.

Email addresses are firstname dot lastname at tufts dot edu, but you can reach all the course staff at once via Piazza at all times.

Instructor Office Hours: Tuesdays, 1:30-2:45pm, and Fridays, 2:30-4pm, or by appointment. In-person office hours will be held in JCC 322. Zoom office hours (any day that we have classes online, or by request) and online appointments will be at my personal Zoom room (see private course page for links).

Course Requirements

Prerequisites: CS 15 and at least one 100-level computer science course, or graduate standing in Computer Science, or permission of the instructor.

No biology background required!

Graduate standing in a related field (Biomedical Engineering, Biology, Genetics) may be sufficient providing your computer programming background is strong enough and you know something about algorithm analysis; check with the instructor, and read the following paragraphs first.

Homework assignments will include several implementation projects in Python. We will learn about algorithms in class and in the readings, but you will then be expected to implement them from scratch and apply them, without much formal help in designing the code.

Also essential will be some basic understanding of algorithm analysis, as is typically covered in CS 15. You should be familiar with asymptotic analysis of algorithmic running times and Big O notation, at least at an introductory level. CS 160 (Algorithms) is helpful but not essential as a prerequisite; material used here will help you when you take Algorithms if you have not yet done so.

Readings: The course textbook is *Understanding Bioinformatics* by Marketa Zvelebil and Jeremy O. Baum, published by Garland Science (a subsidiary of Taylor & Francis Group). Copies of the text should be available in the Medford campus bookstore, or you can order or rent a copy online. Online orders are typically available immediately.
Readings from this text will be listed in the schedule where appropriate. Supplementary readings from the literature or from some of the recommended textbooks listed below appear on the schedule as well.

If you have no biology background, you may want to supplement the readings as well by getting a good introductory molecular biology text. (Several online texts are available for looking up occasional details).

We will do two or three collective "journal club" activities to introduce some class material. These dates are announced on the web schedule. Please read the the journal club papers listed in the schedule *before* class on the indicated day. During class, you will be assigned to join a group of students. Each group will be given a slide with questions on it about some aspect of the paper. You group will collaboratively edit the slide with answers to the questions on that slide. We will then have each team present their slide in order, making up a presentation covering the key points of the whole paper.

Other recommended books:

- *Bioinformatics and Functional Genomics* , by Jonathan Pevsner. A readable introduction to the field. Aimed primarily at biologists, provides somewhat less detail than the course text but may be slightly more approachable.
- *The Cartoon Guide to Genetics* by Larry Gonick and Mark Wheelis. A surprisingly good and serious introduction to the biological concepts covered in this course.
- *An Introduction to Bioinformatics Algorithms*, by N. Jones and P. Pevzner. An algorithms text focusing on examples motivated by computational biology. Helpful if you've never taken an algorithms class; provides a more gentle introduction to selected topics than the following book.
- *Introduction to Algorithms*, by T. Cormen, C. Leiserson, R. Rivest, and C. Stein. The canonical algorithms textbook. Has nothing to do with biology, but should be on every computer scientist's bookshelf.
- *Introduction to Computational Molecular Biology*, by J. Setubal and J. Meidanis. A detailed text focused on computational biology algorithms, aimed at computer scientists. From 1997, but covers several complex topics in depth.
- *Biological Sequence Analysis*, by R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. A good computational biology text focusing on sequence analysis, HMMs, and phylogeny. Includes an excellent whirlwind introduction to statistics.
- *Molecular Biology*, by David Freifelder. A general introductory molecular biology text. Easy to read, a gentle introduction to the topic.
- *Molecular Biology of the Gene*, by J. Watson, N. Hopkins, J. Roberts, J. Steitz, and Alan Weiner. A more advanced and detailed molecular biology text. A very thorough index makes this a good reference book.

Computational resources:

- You will need access to a computer with an internet connection, support for coding in Python, and the ability to remotely log into the department's computer systems.
- **CS account**: The computer science department will provide you with an account on the EECS computer systems for this purpose. **Your LDAP authentication credentials for this account will enable you to log in to the private class materials page.** Thus, this account is essential. Such an account will be automatically provided to all students enrolled in SIS who do not already have one.

  If you need help in obtaining computational resources, you need an account but never received email about its creation, or you are a non-traditional student or auditor who may not be enrolled in SIS, please contact the instructor or teaching assistant as soon as possible. If you didn't receive email but did take a CS course in a prior semester, try **resetting your password** first; most of the time the account still exists and this will work.

  Any code you write for your homework will be graded based on its ability to run on the machine homework.cs.tufts.edu. Please be sure to test your code there; just because it works on your laptop does not mean it will work on a different machine or platform!
- **Gradescope:** You will also need a Gradescope account to submit your work and receive feedback on it. The code for signing up is available on the private class materials page.
- **Piazza:** Finally, there is a class Piazza site linked here that you are encouraged to use to ask questions and discuss topics with your classmates and the course staff. Please take advantage of this resource. You will get faster answers to your questions if you ask the entire group of students and staff at once than if you just email one of us individually.
- We will *not* be using Canvas this semester.

Policies

Grading: Grades will be based on homework assignments (45%), including both written and programming components, two in-class midterms (15% each), a final exam (20%), and class participation (5%).

Late policy: Submissions are due by midnight on the indicated date; Gradescope's timestamp is official. For late work, we are going to use a token policy. You will have 10 tokens for the term. You may use up to 2 tokens per assignment; each token gets you an extra day, which is 24 hours as counted by Gradescope. (**Exception:** homework 4 has a hard 2nd-token deadline of 6pm on April 2nd, to allow for midterm review without compromising the homework.) To use a token, you don't need to tell anyone, just submit and we will count the number of late days as the number of tokens used. **It is your job to keep track of your token usage.** Beyond the 10 tokens, we will not accept late submissions; submit what you have by the deadline for partial credit.

Turning work in on time is important for consistency in grading, because it allows us to discuss homework content in class in a timely fashion. Content builds on previous material, so it is important to figure out quickly if you are lost.

As usual, in the case when your studies are interrupted by serious illness or other truly exceptional circumstances (e.g., situations where your Academic Dean is involved), let us know and we will work something out.

Diversity, Inclusion, and Collegiality: Tufts, the Computer Science Department, and the course staff intend to create a welcoming environment in which all students feel supported and believe that their learning needs and perspectives are valued. We intend to present materials in ways that are respectful to students of any background, ethnicity, race, culture, gender, sexual orientation, or age. We welcome your suggestions on how to improve course effectiveness for yourself or others. If you have religious conflicts with class meetings or requirements, please connect with the course staff.

In this class, we will encourage questions, discussions, and some assignments that involve interacting in groups. While disagreements and differing opinions can be an important part of the learning experience, we expect all students to treat each other with collegiality and respect. Please reach out to course staff if there are any issues with inter-student interactions. While we do not expect this will be necessary, please be reminded that we will, if needed, follow the steps outlined in Tufts' sexual misconduct and non-discrimination policies.

Please also be aware that Tufts faculty are "mandated reporters": if we see, hear, or learn about any kind of discrimination or sexual misconduct, we are required to report it to the university. If you would prefer to access *confidential* counseling for an issue, you can find relevant resources here.

Accomodation for Students with Disabilities: Tufts University values the diversity of our students, staff, and faculty, recognizing the important contribution each student makes to our unique community. Tufts is committed to providing equal access and support to all qualified students through the provision of reasonable accommodations, so that each student may fully participate in the Tufts experience.

If you have a disability that requires reasonable accommodations, please contact the Student Accessibility and Academic Resources (StAAR) Center or call 617-627-4539 to make an appointment with a StAAR representative to determine appropriate accommodations. Please be aware that accommodations cannot be enacted retroactively, making timeliness a critical aspect for their provision.

In addition to following the standard procedures, if you have a disability and would like to discuss how we can better support your learning, please feel free to set up an appointment with course staff.

Academic Integrity: The Tufts academic integrity policy and code of conduct appears here. In particular, plagiarism will not be tolerated. Submitting as your own any written work or code that you did not write yourself, without the help of any other person or entity, is a violation of the academic integrity process.

Please see our collaboration policy below describing what is and is not acceptable in the context of this course. If you are not certain what constitutes plagiarism, please see the academic integrity resources at the link above or ask the course staff.

Please be aware that if Tufts faculty find evidence of academic misconduct, we are required to report it to the university. Penalties can be truly draconian. The time you save in using someone else's work will be lost ten times over as you work through the academic integrity process. So please, don't put yourself through it. We are eager to help you learn what you need to in order to complete complete the work yourself.

Collaboration Policy: *All written work and code submitted should be your own unless you obtain prior permission to collaborate.* You are free to discuss assignments with others in the class unless specifically asked not to, but you must write up your answers and code yourself.

We reserve the right to use computational tools to identify instances of plagiarism or materials (text or code) first written by someone - or something - else, whether published online or previously or concurrently submitted at Tufts. We may make use of plagiarism or similarity detection tools such as TurnItIn, Moss, GPTZero, or other methods to detect inappropriate conduct. We also reserve the right to ask you to verbally explain, in person, any content you submit under your name.

All sources used should be cited. In other words, if you discuss a homework problem with a classmate, you should list that classmate as one of your references for that problem. Please also be warned that not everything you read online is correct. (This is true of print sources as well, but the risk increases greatly online.) Chatbots notoriously hallucinate. Even data from supposedly reputable sources, such as slides posted by faculty at Tufts or other universities, may not have been reviewed by an editor and might contain crucial mistakes. For this reason, I'd like to discourage you from using Google to tackle the problem sets, but if you choose to do so, you must cite the URL(s) that you used. Directly copying text or code from any source without attribution is plagiarism and will be dealt with accordingly.

Course Materials

For homeworks, slides, and other class information, go to the private course materials page. You will need to log in using your CS department account and password. An account will be created for all students registered for the course in SIS who do not already have one.

Tentative Course Schedule:

Updates will occur during the term: check back frequently. Shaded rows refer to past dates.

DATE	TOPICS	READING	OPTIONAL READING
Thurs., Jan. 16	Class overview and administrivia. Introduction to sequences and sequence comparison.	This course Syllabus. Zvelebil & Baum (ZB): Chapter 1 and Section 4.1	For CS students new to biology: Larry Hunter's article, Molecular Biology for Computer Scientists. For bio or BME students or others with less formal CS background: either Corman, Leiserson, Rivest and Stein Chapters 2 + 3, or Jones and Pevzner, Chapter 2: Bio O notation, NP-completeness.
Tues., Jan. 21	Sequence alignment: Global alignment. Dynamic programming. **Hwk 1 out**	This course syllabus ZB: Sections 4.2, 4.5 (pp. 87-89 only); 5.2 (pp. 127-135 only)	Global alignment: Durbin, pp. 17-22.
Thurs., Jan. 23	Local alignment. Scoring schemes, gaps.	ZB: Sections 4.4, 5.2 (pp. 135-140),	Local alignment: Durbin, pp. 23-24, 29-30
Tues., Jan. 28	Scoring matrices, PAM and BLOSUM. **Hwk 1 due**	ZB: Sections 4.3, 5.1	
Thurs., Jan. 30	Database search: introduction, BLAST **Hwk 2 out**	ZB: 4.6-4.7, 5.4	
Tues., Feb. 4	BLAST scoring, Information content; FASTA		Altschul's tutorial on statistics of sequence similarity scores. Altschul's slides on information theory, scoring matrices, and E-values.
Thurs., Feb. 6	Multiple sequence alignment: introduction, star alignment, scoring, NP-completeness **Hwk 2 part 1 due**	Ron Shamir's MSA notes	ZB: 4.5 (pp. 90-93), 6.4-6.5; Durbin, 6.1--6.4
Tues., Feb. 11	MSA iterative and progressive methods	ZB: 6.1	
Thurs., Feb. 13	DNA motifs, profiles. **Hwk 2 part 2 due**	ZB: 6.6	
Tues., Feb. 18	**Midterm 1**		
Thurs., Feb. 20	**NO CLASS**: Monday schedule		
Tues., Feb. 25	Compressive BLAST **journal club**, sublinear search	Compressive BLAST paper	
Thurs., Feb. 27	Gibbs sampling for motif discovery. **Hwk 3 out**	Original paper on the Gibbs sampler for local multiple alignment	Durbin, 6.1--6.4; Original paper on MEME algorithm
Tues., Mar. 4	Sequence assembly: deBruijn graphs; Eulerian paths	The paper about the SOAPdenovo assembler.	
Thurs., Mar. 6	Overlap graphs, Hamiltonian paths, OLC assembly. **Hwk 3 part 1 due**	ZB: 5.3(pp. 141-3)	ARACHNE paper on overlap-based whole genome assembly.
Tues., Mar. 11	Gene finding; Markov models; HMM intro	ZB: 9.2-9.7	
Thurs., Mar. 13	Hidden Markov Models (HMMs) - Viterbi **Hwk 3 part 2 due**	Rabiner handout, pp. 257-266.	
Tues., Mar. 18	**NO CLASS:** Spring Break		
Thurs., Mar. 20	**NO CLASS:** Spring Break		
Mon., Mar. 24	**Hwk 4 out**		
Tues., Mar. 25	Finish Hidden Markov Models; HMM uses in gene finding.		Durbin: chapter 3
Thurs., Mar. 27	EM algorithm **journal club**	short paper on EM algorithms	
Mon., Mar. 31	**Hwk 4 due**		
Tues., Apr. 1	Gene expression: detecting differential expression, multiple testing	ZB: 15.1, 16.1, 16.4	Slonim review article
Thurs., Apr. 3	**Midterm 2**		
Tues., Apr. 8	Gene expression: RNA sequence alignment; clustering and classification **Hwk 5 out**	ZB: 16.2-16.3, 16.5	Golub and Slonim et al., on leukemia classification
Thurs., Apr. 10	Transcriptomic interpretation; functional enrichment		

Tues., Apr. 15	Bioinformatics ethics discussion I **Hwk 5 part 1 due**		
Thurs., Apr. 17	Gene set enrichment analysis **journal club**	Gene Set Enrichment Analysis	
Tues., Apr. 22	Bioinformatics ethics discussion II **Hwk 5 part 2 due**		
Thurs., Apr. 24	Regulatory network inference methods; class wrap-up		
Tues., Apr. 29	Make-up class if needed		
Tues., May 6, 3:30pm-5:30pm	**Final Exam**		

CS 171: Human Computer Interaction (HCI)

Course Number	CS171
Semester	Fall, 2021
Hours	TR 1:30-2:45
Schedule	H+ Block
Location	Alumnae Lounge, Alumnae Hall
Instructor	Remco Chang
Graduate TA	Ellery Buntel Blessing Kolawole
Email	remco at cs_tufts
	Ellery.Buntel at tufts, Blessing.Kolawole at tufts
Office	TBD
Office Hours	TBD
TA Hours	TBD

Course Description

Introduction to human-computer interaction, or how computers communicate with people Methodology for designing and testing user interfaces, interaction styles (command line, menus, graphical user interfaces, virtual reality), interaction techniques (including use of voice, gesture, eye movement), design guidelines, and user interface management software system. Students will design a small user interface, program a prototype, and test the result for usability.

Format: This class meets twice a week (Tuesdays and Thursday). Tuesdays are lecture days, and Thursdsays are studio days. On a studio day selected teams will present their assignment from the prior week. The class will give design and critical feedback. Each assignment is handed out on a Tuesday (after the lecture), and is due the following Tuesday before class. All assignments (with the exception of Assignment 1) will be done in groups. Note that there will be no late submissions. If there are special circumstances tha prevent your team from submitting your assignment on time, you will need to contact the TA or the professor beforehand to receive permission.

Piazza: We'll be using Piazza for class discussions and Canvas for design critique and for submitting assignments.

COVID and Masking Policy: We will follow the university's masking policy strictly. Students (including those who are vacinnated) are required to wear a mask properly during (indoor) class. Failure to comply will result in: (1) the student being marked as disruptive, (2) the lecture ending immediately, and (3) the student being disenrolled from the course.

Schedule

Date	Week	Topic	Due	Out	Notes
9/9/2021	Thursday	Intro: What is HCI (How to submit assignments)		Assignment 1 Out	
9/14/2021	Tuesday	Needfinding and Problem Discovery			(Empathy), (Workarou
9/16/2021	Thursday	Studio 1 -- Practice interviewing skills	Assignment 1 (good/bad design) Due	Assignment 2 Out	
9/21/2021	Tuesday	Studio 2 -- Finding groups			
9/23/2021	Thursday	Problem definition (Intro to WordPress)		Assignment 3 Out	
9/28/2021	Tuesday	Studio 3 -- Practice empthay mapping	Assignment 2 (interview) Due		
9/30/2021	Thursday	Studio 4 -- present interviews			
10/5/2021	Tuesday	Ideate	Assignment 3 (empathy map + POV) Due	Assignment 4 Out	
10/7/2021	Thursday	Solution + Experience Prototypes			
10/12/2021	Tuesday	Video	Assignment 4 (ideation + solution) Due	Assignment 5 and 6 Out	
10/14/2021	Thursday	Studio 5 -- present ideation+solution			

10/19/2021	Tuesday	Design exploration	Assignment 5 (storyboard) Due		
10/21/2021	Thursday	Studio 6 -- present storyboard		Assignment 7 Out	
10/26/2021	Tuesday	Studio 7a -- present videos	Assignment 6 (videos) Due		VIS conferene
10/28/2021	Thursday	Studio 7b -- present videos			VIS conferene
11/2/2021	Tuesday	Visualization and colors			
11/4/2021	Thursday	Studio 8 -- Figma tutorial			
11/9/2021	Tuesday	Design patterns	Assignment 7 (sketches) Due;	Assignments 8 Out	
11/11/2021	Thursday	NO CLASS (Veteran's day)			
11/16/2021	Tuesday	Heuristic evaluation			
11/18/2021	Thursday	Studio 9 -- Run heuristic studies	Assignments 8 (interactive prototype) Due	Assignments 9 Out	Remco at NSF
11/23/2021	Tuesday	NO CLASS			
11/25/2021	Thursday	NO CLASS (Thanksgiving)			
11/30/2021	Tuesday	NO CLASS			Remco out sick
12/2/2021	Thursday	Studio 10 -- share evaluation results	Assignment 9 (evaluation) Due	Assignment 10 Out	
12/7/2021	Tuesday	Human abilities			
12/9/2021	Thursday	Studio 11 -- final project feedback			
12/14/2021	Tuesday	Research and topics in HCI			
12/20/2021	Monday	Final Project Presentation	Assignment 10 (final prototype and presentation) Due		Final exam (H+ Block, 3:30-5:30)

Recommended Reading

- Designing The User Interface: Strategies for Effective Human-Computer Interaction, 6th edition, Shneiderman et al.
- Design of Everyday Things, Don Norman

Grading

Assignments	55%
Studio Work	15%
Website	10%
Critiques	10%
Final Presentation	10%
Total	100%

Assignments:
There are 11 assignments in this class. Each assignment is worth 5% of your final grade. For assignment 1 you will work individually. All other assignments are done in a group (the same group throughout the semester).

Late Policy:
All the assignments due on the designated dates before class. Assignments that are turned in late will not receive credit. Because each of the assignments ties into the next, one late assignment will downstream effects. If you have an extraordinary circumstance, you must contact the instructor or the TA to obtain written approval.

Studio Work:
You will have studio work throughout the semester, including presentations of your work. We will announce the names of the teams when we hand out each assignment. Together, all of your studio work counts for 15% of your final grade.

Website:
You are required to maintain a website for your semester-long project. When you submit your assignment each week, you are also required to update your website with your new work. Updating your website for each of the assignments is worth 1% of your final grade.

Critiques:
In addition to your assignments, each team will be asked to give feedback to another team's work from the week before. As there are 10 assignments, each of your feedback will be worth 1% of your final grade.

Final Project:
You will be asked to present your project at the end of the semester. Your presentation of your final project is worth 10% of your final grade.

Accommodation

Tufts is committed to providing support services and reasonable accommodations to all students with documented disabilities. To request an accommodation, you must register with the Student Accessibility Services at the beginning of the semester.

Acknowledgement

Course lecture and material are based on the HCI course at Stanford by Professor James Landay.

CS 11 - Spring 2025

ow you will find our **proposed** schedule for the semester. Topics and assignments are subject to change, and may lated as the semester progresses.

eek 1: 1/12/2025

Wednesday	**Lab 0:** Linux or Bust
Thursday	**Lecture:** Welcome - Introduction (pre-recorded lecture videos) Read: CS11 Syllabus encrypt.cpp **HW0 Out:** Holy Shift

eek 2: 1/19/2025

Monday	No School - MLK Day
Tuesday	**Lecture:** Variables, Data Types, and Arrays (pre-recorded lecture videos) Read: Sec. 1.3, 2.1-2.3, 7.1 temperature.cpp temp_float.cpp phone_number.cpp
Wednesday	No Lab - Fake Monday
Thursday	**Lecture:** Conditionals and Boolean Expressions (pre-recorded lecture videos) Read: Sec. 2.4 scope.cpp

eek 3: 1/26/2025

Tuesday	**Lecture:** Loops (pre-recorded lecture videos) Read: pp. 84-91, Sec. 3.3-3.4 **HW0 Due** by midnight
Wednesday	**Lab 1:** What the Diff? **HW1 Out:** Lock and Load
Thursday	**Lecture:** Functions (pre-recorded lecture videos) Read: Sec. 4.1-4.4, 7.2 arg_test.cpp functions.cpp

eek 4: 2/2/2025

Tuesday	**Lecture:** File I/O (pre-recorded lecture videos) Read: Sec. 6.1, pp. 355-356 file_basics.cpp numbers.txt pitfalls.cpp words.txt eof.cpp integers.txt **HW1 Due** by midnight
Wednesday	**Lab 2:** Crash Test **HW2 Out:** Snow Crash
Thursday	**Lecture:** Strategy and Debugging (pre-recorded lecture videos) bowling.cpp bowling_test.in

eek 5: 2/9/2025

Tuesday	**Lecture:** Memory Part 1: Pointers (pre-recorded lecture videos) Read: pp. 510-514 (i.e., Ch. 9.1 until they start talking about 'new') **HW2 Due** by midnight
Wednesday	**Lab 3:** Baby G **HW3 Out:** Tradecraft
Thursday	**Lecture:** Memory Part 2: Heap Allocations (pre-recorded lecture videos) Read: The rest of Ch 9.1, Ch. 9.2 grade_bot.cpp

eek 6: 2/16/2025

Monday	No School - Presidents' Day
Tuesday	**Lecture:** Structs (pre-recorded lecture videos) Read: Sec. 10.1 celebs_complete.cpp celebs_start.cpp celebs.txt **HW3 Due** by midnight
Wednesday	**Lab 4:** Ye Olde Strings **HW4 Out:** Word Play
Thursday	No Class - Fake Monday

eek 7: 2/23/2025

Tuesday	**Lecture:** Memory Pt. 3: Putting it all together (pre-recorded lecture videos) partners.cpp students.txt **HW4 Due** by midnight
Wednesday	**Lab 5:** Black Book **HW5 Out:** Phone Tree
Thursday	No Class

eek 8: 3/2/2025

Tuesday	**Lecture:** Monster Mash: More Pointers in Structs **HW5 Due** by midnight
Wednesday	**Lab 6:** Query Quaziness **HW6 Out:** Mutations
Thursday	**Lecture:** Complexity and Randomness Read: pp. 994-997 (i.e., the part of Sec. 18.3 labeled "Running Times and Big-O Notation") Read: pp. 188-189 (i.e., the part of Sec. 4.2 labeled "Random Number Generation")

eek 9: 3/9/2025

Tuesday	**Lecture:** Midterm Review **HW6 Due** by midnight
Wednesday	**Lab 7:** Midterm Review **HW7 Out:** Airtight
Thursday	**Midterm Exam** In Class

eek 10: 3/16/2025

Monday	Spring Break!
Tuesday	Spring Break!
Wednesday	Spring Break!
Thursday	Spring Break!
Friday	Spring Break!

eek 11: 3/23/2025

Tuesday	**Lecture:** Recursion (Part 1) Read: Ch. 14 **HW7 Due** by midnight
Wednesday	**Lab 8:** Recurse, Recurse, Recurse, Recurse... **HW8 Out:** Lineage
Thursday	**Lecture:** Recursion (Part 2)

eek 12: 3/30/2025

Tuesday	**Lecture:** Object-Oriented Programming: Intro Read: Sec. 6.1 (Only the part titled "Introduction to Classes and Objects"), Sec. 10.2 (Up through th part titled "Summary of Some Properties of Classes") **HW8 Due** by midnight
Wednesday	**Lab 9:** Fetch **HW9 Out:** Advising Period
Thursday	**Lecture:** Object-Oriented Programming: Next Level Read: Sec. 10.2 (the rest of it), 11.4 (Stop when you hit "Copy Constructors") What goes in a .h and what goes in a .cpp?

eek 13: 4/6/2025

Tuesday	**Lecture:** OOP Implementation: Vectors Part 1 **HW9 Due** by midnight
Wednesday	**Lab 10:** Aces **HW10 Out:** Rack-O
Thursday	**Lecture:** OOP Implementation: Vectors Part 2

eek 14: 4/13/2025

Tuesday	**Lecture:** Inheritance **HW10 Due** by midnight
Wednesday	**Lab 11:** Quell the Compiler **Final Project Out:** Sushi Go!
Thursday	No Class - Work on Project

eek 15: 4/20/2025

Tuesday	No Class - Work on Project
Wednesday	**Lab 12:** Project Work (Optional)
Thursday	No Class - Work on Project

eek 16: 4/27/2025

Monday	Classes End **Final Project Due** by midnight
Tuesday	Reading Period
Wednesday	Reading Period
Thursday	Reading Period

eek 17: 5/4/2025

Tuesday	**Final Exam**

CS 11 - Spring 2025

HOME SYLLABUS SCHEDULE CODING STYLE GUIDE CS11 TECH GUIDE

tructor: Elyse Cornwall
ail: elyse.cornwall@
ice: Joyce Cummings Center, Room 457
ice Hours: Tues 2:30-4pm

Lecture: Joyce Cummings Center, Room 270
Section 1: Tues/Thurs 10:30-11:45am
Section 2: Tues/Thurs 12:00-1:15pm
Lab: Joyce Cummings Center, Rooms 235 and 240 (see

lterm Exam: This course will have an in-person midterm exam in class on Thursday, March 13th.
al Exam: This course will have an in-person final exam on Tuesday, May 6th. In order to take CS11, you must be ab end this final exam. The Section 1 exam is from 3:30-5:30pm. The Section 2 exam is from 12:00-2:00pm.

ble of Contents

s page contains course information pertaining to:

ourse Overview

s course serves as an introduction to computer science via the programming language C++. You will learn how to vise precise procedures for solving problems, and how to specify these procedures using the C++ programming guage. Along the way, you will strengthen your computational thinking skills (helpful for any form of problem solvin l begin to form a mental model of how a computer operates.

11 is a fast-paced, challenging course that may require more of your time and effort than what you've experienced er courses. By staying aware of the course policies, following the advice of the course staff, and taking advantage provided resources, you will set yourself up for success in this course regardless of your background.

ourse Goals

the end of this course, students should be able to:

1. Demonstrate computational problem solving with basic programming constructs.
2. Interpret a sequence of English instructions (an algorithm) as a C++ program and vice versa.
3. Apply computational thinking to solve problems.
4. Assess a C++ program's aesthetic value and functional correctness.

ourse Expectations

ese expectations are designed to prevent students from missing important information or misunderstanding policie
t are essential for succeeding in the course. If you experience a negative outcome in CS11 (e.g. losing points on an
essment, missing a regrade request window, not begin granted an extension) due to a failure to meet one of these
pectations, that outcome will not be changed.

a CS11 student, you agree to:

- Read this entire syllabus during the first week of class.
- Adhere to all the policies outlined in this syllabus.
- Read all pinned Piazza posts carefully throughout the semester (Piazza is explained below). This is the main mechanism for distributing course information such as changes to policies, deadlines, etc.
- Follow the three CS 11 axioms:

    1. **Start early.** Unlike a set of individual problems, a program has to function as a whole. The more you build, the more carefully you must think about how the various components fit together. If you're in a rush, it will be difficult to build a functional program; give yourself as much time as possible!
    2. **Think before you code.** A program is just a faster and more reliable version of a procedure that can be done with a pencil and paper. If you do not have a clear idea of how you would solve the problem with a pencil and paper, then you are not ready to write code for it.
    3. **Write a little, test a little.** Finding a bug in 5 lines of code is far easier than finding a bug in 500 lines of code. A good program is written in small, testable increments. You should not write the next section of your program unless you have a clear plan for how to test it.

nal Grade Breakdown

ur course grade will be produced as a weighted sum of your scores in four categories:

1. Labs (10%)
2. Homework (60%)
3. Midterm Exam (10%)
4. Final Exam (20%)

ades will be posted on Gradescope for students to view. We do not expect to apply a curve to any portion of this gr

omework Logistics

general, homwork assignments will go out every Wednesday and will be due at midnight the following Tuesday (the
be a longer project towards the end of the semester). Each homework assignment will consist of three componen

1. A written component that will be submitted via Gradescope
2. A programming component that will be submitted via our **`submit11`** system
3. A style component that evaluates how well your code conforms to our style guide

mework grades will be posted to Gradescope for students to view.

ritten

e written component of your homework must be submitted as a PDF via Gradescope. Failing to submit your written
nponent or submitting a non-PDF document will result in a loss of credit for this component.

ogramming

e programming component of each assignment must be submitted via our `submit11` system. First and foremost, t
graming component of your homework **must compile** on our Halligan servers in order to receive any credit. Once
omission has compiled, it is evaluated by comparing its output to the output that our solution code produces.

h the exception of HW0, programming submissions will be graded by an automated system that uses the program
`ff` to compare a submission's output to our ground truth. Students will learn to use `diff` during their second lab
sion and are expected to use it for testing before submitting all of their subsequent homework assignments.

the majority of the homework assignments, we will be providing one or more sample input files that students may
testing. While these tests are designed to be generally representative of the tests we will be using for grading, the
intended to be comprehensive. Students are expected to create their own input scenarios and test them using `di`

e triplicate of **compile-diff-submit** should become the cornerstone of every student's submission process.

te Homework

e Late Token System

recognize that in some circumstances, it is not possible to meet an assignment deadline. In these circumstances,
n utilize the late token system. Each student is automatically issued **five** “late tokens” to be used on homework
ignments; a maximum of two tokens can be used on a single assignment. A late token grants you a 24-hour extens
an assignment, and requires no action on your part; our grading software will automatically check the date and tim
ır submission and deduct the appropriate number of tokens. Note the following late token rules:

1. Once you are out of late tokens, late homework will receive no credit.
2. If you don't want to use a token, don't submit anything after the due date. We grade your most recently submitted work!

e total number of tokens used on an assignment is reported along with your homework and lab grades on Gradesc
fortunately, Gradescope does not provide a summary of how many late tokens you’ve used over the semester; it is
ponsibility to keep track of your late token usage.

e tokens are designed to accommodate short-term setbacks like catching a cold or an ill-timed deadline in anothe
ss. Again, there is no need for any emails or explanations. Just turn in the assignment when you get it done, and th
e token accounting will happen automatically.

an-Approved Extensions

nore extreme extenuating circumstances, the late token system may not be enough. In these cases, students must
actively reach out to their dean to request an extension (note that we rarely grant extensions that are requested or
t the due date). Your dean will then work with the course instructor to make appropriate arrangements.

grade Requests

ı are always welcome and encouraged to ask for an explanation of the grade you received; simply post on Piazza
vately. If you want to request a regrade for an objective grading error on our part, you must submit a request via

ıdescope within **one week** of the assignment grade being released. If you're unsure how to submit a regrade reque
a particular component of an assignment, it's fine to submit the request on the first problem of the assignment or
ır request as a private Piazza post.

egrade request may or may not result in a new grade being assigned. In some cases, a regrade request will be grar
incur a point penalty. If we have to make a small manual change to your code to regrade your assignment success
ır grade for that component will take a 10 point penalty (so your max grade would be a 90%). If this change could l
en avoided by running the `diff` program appropriately before submitting the assignment, a 20 point penalty will be
plied (so your max grade would be an 80%). Finally, since written components are worth 10 points themselves, we
accept regrade requests for written components that would require uploading a new PDF to Gradescope.

b Logistics

› attendance is mandatory. Our weekly labs are designed to prepare students for the current homework. In lab,
dents will work with a partner to complete a small problem or activity in the allotted time.

ally, you will finish the lab in the allotted time, but it's totally ok if you don't. Simply submit what you have at the en
ır lab session and, if you feel comfortable with the concepts involved, move on to your weekly homework assignme
ı're also welcome to keep working on the lab after your lab session, and bring lingering questions to office hours.

›s are due by the Friday of the week the lab was released. As long as you've put in a good faith effort, you will rece
nplete (100%) lab grade. Lab grades will be posted to Gradescope as a component of the weekly homework grade
ıere to the same regrade deadlines described above.

chnical Resources

xtbook

s course has one optional textbook, which serves as a helpful supplementary reference: Problem Solving with C++
lter Savitch. ISBN: 0133591743 (7th Edition or higher).

ızza

e preferred means of contacting the course staff is via Piazza, an online forum where students can ask and answer
estions. General questions about the homework, course policies, C++, etc. should be posted publicly so that your
ssmates can benefit from the answers and any resulting discussion. Before posting a question, please check to se
ether your question has already been asked. Questions that are personal in nature, or that pertain to specific piece
le that you have written should be posted privately to the course staff.

**zza will host all of our major course announcements. It is your responsibility to check it regularly to avoid
ssing any information.**

ch Guide

ı'll rely on a small number of programs and scripts to complete and submit labs and homework assignments. It is
cial that you arrive at a basic understanding of these tools in order to complete your work without issues. You will l
out these tools during the first few weeks of class, but further descriptions and instructions for them can be found
CS11 Tech Guide.

s and Office Hours

s course is challenging, but we want to help you succeed! If you need help understanding a concept, tackling an ignment, or dealing with a bug, you can participate in our teaching assistant (TA) office hours (OH) sessions, whic e place in the large room overlooking the baseball fields on the 4th floor of the Joyce Cummings Center, Room 40 en you arrive, you will "join the queue" by writing your name on a list maintained on a whiteboard. If you plan to wa newhere other than the OH room (which must be somewhere nearby), write your location next to your name and a come find you when it's your turn. The full **OH schedule** is posted and kept up to date on a pinned Piazza post.

assistance is meant to get you un-stuck or provide a nudge in the right direction; we expect that you will try to gra n your issue yourself before asking for help. TAs may turn away students who cannot provide evidence of attempti ve their problem on their own. For example, "What does this compiler error mean?" or "I don't understand this part assignment's directions."

: of respect for our TAs' time, office hours will end at the posted time regardless of the length of the remaining que e queue will be longer closer to the assignment deadline, so don't assume that you'll receive help if you join the que ing the last 30 minutes of office hours on a Tuesday night.

ollaboration and Academic Integrity

ı are encouraged to discuss general CS11 concepts with *anyone*. This includes lecture slides and coding demos, la ndouts, reading material, and C++ concepts (functions, pointers, classes, etc.) and syntax ("how do I write a for loo w do I create an integer variable?"). You are also welcome and encouraged to fully collaborate on labs with a partn

e collaboration policy for homework assignments is more nuanced. As described above, homework assignments w nprised of two components, a written component and a programming component. For each of them, we have sligh erent policies about which forms of outside help and collaboration are acceptable:

itten

e goal of our written problems is to give you practice finding pieces of information that you have not expressly bee en in class and lab. This skill allows you to eventually function independently as a programmer, and is critical for all nputer scientists to master. Thus, you are expected to find the answers to written problems yourself. You may **not** ıfer with classmates, and may only minimally query the TA staff. However, any means of finding an answer is fair ga ı may search the internet, consult the textbook, write test programs - anything.

ogramming

h your classmates and the internet can be invaluable resources when programming, so long as they are used cording to the following policies:

1. You may search the internet and talk to other students about general programing concepts and C++ syntax, but **not** about assignment-specific code or strategies. For example, it is fine to ask "How do you structure a 'while' loop in C++?", but it is **not** fine to ask "How do you write a Caesar Cipher decryptor in C++?" Assignment-specific questions should only be posed to the course staff.
2. You should never be looking at or discussing another student's code and no one outside of our course staff should be looking at or discussing your code. This also applies to testing and debugging: you may not help test or debug another student's code, and you may not receive help testing or debugging your code from

anyone other than the current course staff. If a TA sees you participating in any of these activities, it will be reported.

3. Only submit code that you can explain. We reserve the right at any time to ask you to explain a piece of code that you submitted. If you cannot explain the code, then we will have no choice but to assume that it is not your work.
4. Do not plagiarize code! Lifting partial or complete solutions from anyone (classmates, online sources, strangers) is completely prohibited. You must not submit code that others wrote or that you wrote for a previous class (or a previous iteration of CS 11).
5. No questions, student solutions, or instructor-provided solutions should be posted online in any capacity (except as a submission to Gradescope). This means that you are prohibited from posting any of your work for CS11 in a public Github repository.

y violation of the above policies will be considered cheating, and all students involved in any capacity will be warded directly to the Office of Student Affairs, who will investigate the case independently. Their sanctions ge from horrible to inconceivably horrible. It's not worth it.

clusivity and Accessibility

spect is demanded at all times throughout the course. We realize that everyone comes from a different backgroun n different experiences and abilities. In the classroom, participation is encouraged, and should always be used to b ryone in the class.

ts University values the diversity of our students, staff, and faculty, recognizing the important contribution each dent makes to our unique community. Tufts is committed to providing equal access and support to all qualified dents through the provision of reasonable accommodations so that each student may fully participate in the Tufts perience. If you have a disability that requires reasonable accommodations, please contact the StAAR Center to ermine appropriate accommodations. Please be aware that **accommodations cannot be enacted retroactively**, king timeliness a critical aspect for their provision.

CS 15 - Spring 2025

Note: All lectures will be recorded and posted on the course Canvas page under "Echo360."

Week 1: 1/12/2025

Tuesday	**No lab held** **Lab:** (Optional) Quick Intro to Writing, Running, Submitting Programs
Wednesday	**Lecture:** Welcome - Introduction Read: CS15 Admin & Policies Browse: CS15 Reference Page Read: Shaffer 1.1-1.2 **Lecture:** C++ Review: Classes (and more!) employee code **Lecture:** ArrayLists IntArrayList.h vectors.cpp Read: Intro to Valgrind
Thursday	
Friday	

Week 2: 1/19/2025

Monday	No School
Tuesday	**Lab:** Unit Testing, ArrayLists **HW1 Out:** ArrayLists
Wednesday	**Substitute Monday Schedule** **Lecture:** The Big Three shallow_example.cpp deep_example.cpp **Lecture:** (Virtual Mini-Lecture) Exceptions Slides exceptions.cpp Read: CS 15 Style Guide
Thursday	
Friday	

Week 3: 1/26/2025

Monday	**Lecture:** Linked Lists 1 StringLinkedList.h Read: Shaffer 4.1.2, 4.1.5
Tuesday	**Lab:** Linked Lists **HW1 due by 23:59:59** **HW2 Out:** Linked Lists
Wednesday	**Lecture:** Linked Lists 2 **Lecture:** Makefiles make example code Read: Makefile Handout
Thursday	
Friday	

Week 4: 2/2/2025

Monday	**Lecture:** Complexity Read: Shaffer 3.0-3.5
Tuesday	**Lab:** make, diff **HW2 due by 23:59:59** **Project 1 Out:** MetroSim
Wednesday	**Lecture:** Queues Read: Shaffer 4.3 **Lecture:** Stacks paren_matching.cpp Read: Shaffer 4.2
Thursday	
Friday	**Lab:** (prelab) Circular buffers

Week 5: 2/9/2025

Monday	**Lecture:** File I/O isprime.cpp Read: File I/O Handout **Lecture:** Recursion
Tuesday	**Lab:** Stacks, Queues, and Circular Buffers
Wednesday	**Proj 1 checkoff and phase 1 due by 11:59pm** **Lecture:** Binary Search
Thursday	
Friday	

Week 6: 2/16/2025

Monday	No School
Tuesday	**Lab:** Recursion **Project 1 due by 23:59:59** **Project 2 Out:** CalcYouLater
Wednesday	**Lecture:** Project 2 Background **Lecture:** Trees int_tree_example.cpp
Thursday	**Substitute Monday Schedule: WE HAVE LECTURE!** **Lecture:** Binary Trees and their Traversals **Lecture:** Sets
Friday	

Week 7: 2/23/2025

Monday	**Lecture:** Binary Search Trees (BSTs)
Tuesday	**Lab:** Binary Tree Traversals
Wednesday	**Proj 2 design checkoff and phase 1 due by 11:59pm** **Lecture:** BST (cont'd) and AVL Trees Lecture Recording Read: Shaffer 13.2-13.2.1
Thursday	
Friday	**Lab:** (prelab) AVL Trees Preview

Week 8: 3/2/2025

Monday	**Lecture:** AVL (continued) **Lecture:** Templates
Tuesday	**Lab:** AVL Trees **Project 2 Due by 23:59:59**
Wednesday	**Lecture:** Midterm Info **Lecture:** Huffman Coding
Thursday	
Friday	

Week 9: 3/9/2025

Monday	**Lecture:** Midterm Review
Tuesday	**Lab:** Midterm Review (no spec)
Wednesday	Midterm
Thursday	
Friday	

Week 10: 3/16/2025

Monday	No School
Tuesday	No School
Wednesday	No School
Thursday	No School
Friday	No School

Week 11: 3/23/2025

Monday	**Lecture:** Priority Queues and Heaps **Project 3 Out:** zap
Tuesday	**Lab:** Heaps
Wednesday	**Lecture:** Hashes
Thursday	
Friday	

Week 12: 3/30/2025

Monday	**Lecture:** Hashes (continued)
Tuesday	**Proj 3 phase 1 due by 11:59pm** **Lab:** Hashes
Wednesday	**Lecture:** Intro to Graphs
Thursday	
Friday	

Week 13: 4/6/2025

Monday	**Lecture:** Graph Traversals
Tuesday	**Lab:** Graph Traversals **Project 3 due by 23:59:59** **Project 4 Out:** gerp
Wednesday	**Lecture:** Dijkstra's Algorithm
Thursday	
Friday	

Week 14: 4/13/2025

Monday	**Lecture:** Sorting I
Tuesday	**Lab:** Dijkstra's Algorithm Worksheet

Wednesday	**Proj 4 design checkoff and phase 1 due by 11:59pm** **Lecture:** Sorting II
Thursday	
Friday	

Week 15: 4/20/2025

Monday	No School
Tuesday	**Lab:** Sorting
Wednesday	**Lecture:** Sorting: Non-Comparison Sorts **Project 4 due by 23:59:59**
Thursday	
Friday	
Saturday	

Week 16: 4/27/2025

Monday	**Lecture:** Final Info **Lecture:** Wrap Up **Lecture:** CS 40 Preview
Tuesday	**Reading Period**
Wednesday	**Reading Period**
Thursday	**Reading Period**
Friday	**Final Exam (May 2nd)** Sec. 1: 12-2pm Sec. 2: 3:30-5:30pm

CS 105: Programming Languages

Spring 2025

https://www.cs.tufts.edu/comp/105

Class Sessions	Monday, Wednesday 3:00pm–4:15pm JCC 270
Instructor	Richard Townsend, richard.townsend@tufts.edu
Richard's Office Hours (JCC 440A)	Tuesdays 2:00pm-4:00pm (or by appointment)
Final Exam	Friday, May 2, 3:30pm-5:30pm, JCC 270

Course Overview

Summary

CS 105 introduces you—through extensive practice—to ideas and techniques that are found everywhere in today's programming languages. You will learn high-level, flexible programming skills that are applicable to older languages, popular modern languages, and even languages that don't yet exist. No matter what language you work in, when you finish 105, you'll be writing more powerful programs using less code. At the same time, you will learn the mathematical foundations needed to talk precisely about languages and programs: abstract syntax, formal semantics, and type systems.

You will explore and apply programming language concepts through multiple case studies, conducted using (mostly) tiny languages that are designed to help you learn. In any given case study, you may act as a practitioner (by writing code in a language), as an implementor (by working on an interpreter for the language), as a designer (by inventing semantics for a related language), or as a scholar (by proving mathematical properties of the language).

Logistics

The main point of access for the course is the website (the link is at the top of this document): it provides all the slides, readings, homeworks, solutions, and recitation materials.

Here is the weekly flow of the course:

1. Come to class sessions: take notes, participate in partner-based activities, and ask questions! This will ensure you're prepared for assignments and recitations. **Recitations will not be useful to you if you haven't come to class first.**

2. Read over the homework spec *before* attending recitation. Ask questions about it on Piazza and in office hours (OH), or talk about it with your classmates. Start the assigned reading.

3. Go to recitation, where you will get supervised practice working on problems that resemble the homework problems.

4. Work on your assignment, first by doing the reading and comprehension questions and then by completing the programming and/or proof problems. Make sure you spread this out so you have breaks between your 105 work. Post on piazza or come to OH if you need assistance.

If you feel like you're falling behind or are overwhelmed with 105, please reach out to a member of the course staff; we want to help you!

Prerequisites

- CS 15 Data Structures: You need substantial programming experience to keep up with the homework, particularly working with dynamically allocated (i.e., heap-allocated) data structures and recursive functions. We will also be referencing many classic data structures (stacks, queues, lists, trees) and algorithms (sorts, graph algorithms) throughout the course.
- CS/MATH 61 Discrete Math: You need some experience proving theorems, especially by induction. You should also be comfortable reading and writing formal mathematical notation.

Course Goals

By the end of this course, students should be able to

1. Design code using algebraic laws.
2. Read and write code that uses functional programming techniques.
3. Recognize the merits of polymorphism, type checking, and type inference.
4. Use functional and object-oriented language features to hide implementation details.
5. Understand precise specifications of how programming languages work.
6. Mathematically prove the behavior of programs written in a given language.

Technical Resources

Office Hours

This course is challenging, but we want to help you succeed! If you need help understanding a concept or tackling an assignment, or dealing with a pernicious bug, you can participate in our TA office hours sessions. You can also attend an instructor's office hours for any of these reasons, to discuss any issues you're having with the course, or just to say "hi!"

TA office hours will take place in the middle of the main hallway on the 3rd floor of the Cummings Center (near the stairs). When you want help, write your name and your place in the queue (as a number next to your name) on the whiteboard.

Office hour assistance is meant to help you along or provide a nudge in the right direction; we expect that you will try to grapple with your issue yourself before asking for help. To help you follow this practice, you are expected to provide a specific topic or question to receive help. Here are some *poor* examples of specific topics/questions: "why doesn't my program work?", "debugging", "can you explain impcore?". Here are *much better* examples: "why does my function produce a number when it should produce a list?", "debugging a syntax error", "how do I call a function in impcore?".

The full office hours schedule is posted and kept up to date as a pinned Piazza post; make sure you check it before trekking to Cummings!

Piazza

We will be using Piazza for class discussion; you can get to the 105 page via this link or on the Home page of the course website. **Piazza will host all of our major course announcements; it is your responsibility to check in regularly to avoid missing any important information.**

In general, you should post all questions related to the course on Piazza; post publicly if at all possible, as other students may have similar questions or be able to help you faster than the course staff. Please post privately to the course staff if your question reveals individual work (e.g., algebraic laws or code) or concerns your grades.

Textbooks

This course has two required textbooks:

- *Programming Languages: Build, Prove, and Compare.* Norman Ramsey. Tufts University, Fall 2022.

  You must have this specific edition; if you're looking at an old version it may no longer align with this semester's assignments! The correct edition can be purchased from the Tufts Bookstore. If paying for this book is a financial hardship or you have any other questions about obtaining the textbook, please contact Sarah White (she's great) at s.white@tufts.edu.

- *Seven Lessons in Program Design.* Norman Ramsey.

  This short booklet is linked from our course website (at the bottom of the home page).

There is also one optional textbook that is useful for the few ML assignments we have:

- *Elements of ML Programming, ML97 Edition.* Jeffrey Ullman. Pearson, 1997.

Tentative Schedule

The tentative schedule for the course is displayed on our course website. Topics and assignments are subject to change, and may be updated as the semester progresses (you will be informed of any changes as soon as they happen). Typically: Lectures will be on Mondays and Wednesdays; homeworks will be due on Tuesdays at 11:59pm and new homeworks will go out Wednesday mornings; and recitations are either on Thursday or Friday (depending on which recitation you are enrolled in). **There are exceptions to all of these throughout the semester** (e.g., sometimes homework will be due on different days, etc.). Therefore, it is important for you to keep up with the schedule linked above.

Assessments and Grading

Your course grade will be produced as a weighted sum of your scores in four categories:

1. Recitations (5%)
2. Homework: Comprehension Questions (5%)
3. Homework: Programming and Proof Problems (70%)
4. Final Exam (20%)

All grades will be posted on Gradescope for students to review.

Although your ultimate course grade will be a conventional letter, much of your specific work will be graded on a scale of No Credit, Poor, Fair, Good, Very Good, and Excellent. These grades correspond to numeric values of 0, 65, 75, 85, 95, and 100, respectively. For example, if an assignment has 3 equally weighted components and you received a Good on two parts and a Very Good on the third, your grade for that assignment would be approximately .33 * 85 + .33 * 85 + .33 * 95 = 87%. In a typical class, a consistent record of Very Good work will lead to a course grade in the A range. Work rated Good corresponds to a wide range of passing grades centered around B. Work rated Fair will lead to low but satisfactory course grades around a C; if a significant fraction of your work is Poor, you can expect an unsatisfactory grade.

Recitations

Recitations are graded as Attended or Not Attended (a recitation TA will take attendance in each session); your final recitation grade is the percentage of recitations you attended. We drop your two lowest recitation grades, effectively letting you miss two recitations at no penalty. There is no such thing as an “excused absence” from recitation, so if you need to miss one just let your recitation leader know. If you need to miss more than two recitations due to a personal emergency, please explain the situation to your academic dean and ask them to contact a course instructor.

Homework

In general, homeworks will go out every Wednesday and be due at 11:59:00PM EST the following Tuesday (there will be a few larger assignments in the second half of the course that are exceptions to this rule). Each homework will have two components:

1. Your responses to a set of reading comprehension questions (CQs), which will be submitted as a .txt file.

2. A programming and/or proof component.

You will submit all of your work directly to Gradescope. No submissions will be accepted via email or other means. To avoid missing the strict 11:59:00PM EST deadline, you should submit whatever you have a bit earlier (or much earlier!) even if incomplete; you may submit as many times as you wish before the deadline. We will always grade the latest submission.

After submitting to Gradescope, you must always look over the files you submitted to make sure that they are what you expect. In particular, make sure: (1) you submitted all necessary files; (2) the code/files you submitted are the latest versions of your work; and (3) any PDFs you submitted are readable and not corrupted. (3) is important, since VSCode has been known to corrupt PDFs that are synced with the department servers. **It is your responsibility to check that the files you submitted are the correct and readable versions. Regrade requests will not be considered for submitting incorrect files.**

Comprehension Question Grades A set of comprehension questions (CQs) is associated with each homework assignment. Your answers to these questions are graded based on the number of questions answered and the correctness of each answer, roughly following this scale:

- **Excellent**: All answers are completely correct.
- **Very good**: All answers are mostly correct.
- **Good**: A majority of answers are mostly correct.
- **Fair**: Most answers are incorrect, but all were reasonably attempted.
- **Poor**: Most answers are incorrect or blank.
- **No Credit**: No answers are provided.

Programming and Proof Problem Grades The programming and proof problems on each assignment will be graded both for correctness (e.g., Do your programs pass our automated tests? Are your theoretical proofs and rules accurate and complete?) and for structure and organization (e.g., Does your code follow our course coding standards? Are your algebraic laws accurate and complete?). Correctness is typically evaluated by automated testing scripts that list which tests you passed or failed and why. Structure and organization is evaluated manually by our course staff. Both of these components are graded with a coarse five-point scale:

- **Excellent**: Your submission is outstanding in all respects.
- **Very Good**: Your submission does everything asked for, and does it well. There may be something small that could be improved.
- **Good**: Your submission demonstrates quality and significant learning.
- **Fair**: Your submission is quite lacking in one or more aspects; key issues need to be addressed. This is the lowest satisfactory grade.

- **Poor**: Your submission shows little evidence of effort or has other serious deficiencies. This is an unsatisfactory grade.
- **No Credit**: No submission provided OR the submission exhibits plagiarism OR the submission violates a rule that the homework indicated would result in no credit.

Late Policy

For every assignment we have, you are allowed to submit up to 24 hours after the posted deadline. There is no penalty for submitting within this 24-hour grace period. However, typically a new assignment will be released immediately after each deadline—so by submitting late, you lose some time on the next assignment. If you submit more than 24 hours after a deadline, your submission will not be accepted.

You should think of these late submissions as extensions you have been granted ahead of time and use them when you might have otherwise tried to ask for an extension. There are no additional "late tokens" in this course. You can think of our late policy as granting one "late token" per assignment.

If you experience an extraordinary difficulty, such as serious illness, family emergencies, or other extraordinary unpleasant events, your first step should be to contact your advising dean as soon as you can: explain the situation to them and ask them to contact a course instructor. Your dean will work with the instructor to make appropriate arrangements. **IMPORTANT: The earlier you notify your dean, the more flexibility the course staff will have to make appropriate arrangements.**

Regrades and Grade Explanations

If you want to request a regrade for an objective error on our part, you must submit a private request to the Instructors via Piazza; provide your UTLN and explain the error. The deadline for these requests is **one week** from the time the grades were posted for a given assignment; no exceptions to this policy will be made. A regrade request may or may not result in a new grade being assigned.

If a regrade request requires the course staff to make a manual modification to your submission, your grade for the component being regraded will be capped at a "Good" as a penalty. **IMPORTANT: If you misname a function, you could face a significant penalty from the autograder. Therefore it is crucial that you check that all function names match what the spec calls for exactly.**

You are always welcome to ask for an explanation of the grade you received. We want to help you understand how your work was evaluated and how you can continually improve in the course!

Collaboration and Academic Honesty

If you have **any** questions about the below policies or a specific situation dealing with academic integrity, do not hesitate to ask a course instructor. All students are expected to read and adhere to the Tufts Academic Integrity Policy.

Collaboration

In general, you are encouraged to discuss the lecture content, reading material, and what's being asked of you on assignments with your classmates. However, **you must not discuss anything that you produce for an assignment with classmates in any form (e.g., English, pseudocode, pictures, etc.).** Examples include proofs, code, algebraic laws, and function contracts. You may not show your work at this level to other students nor may you look at others' work at this level. This also applies to testing and debugging: you may not test or debug another student's code or let them test or debug your code. Due to this policy, any Piazza post that discloses your work on an assignment must be posted privately.

Lifting or looking at partial or complete solutions from anyone (classmates, online sources, strangers) is completely prohibited. You may not work on the specific problems with anyone other than TAs or a course instructor, and you should not use the internet in any capacity to help solve any specific problem. No homework problems, student solutions, or instructor-provided solutions should be posted online in any capacity (except as a submission to Gradescope); **this means that you are prohibited from posting any of your work for 105 in a public Github repository**.

If an assignment contains a *pair programming* component and you choose to pair program, you are welcome to discuss/view any part of that component with your partner, but not any other students.

Finally, you **may not** use ChatGPT, Copilot, or similar large language models to assist in completing your homework. Our assignments exist for your benefit—by relying on AI to do your work for you, you only waste your own opportunity to learn.

Policies for Students Retaking CS 105

If you are repeating the course or any part of it (e.g., you withdrew or dropped after doing one or more assignments), you must let a course instructor know at the beginning of the semester. In this case, you are expected to abide by the following policies:

- There is no acceptable use of instructor-provided solutions from prior semesters. If you have kept any such solutions, destroy them.
- If you have written your own solutions for past semesters, it is acceptable to consult them for ideas, and it is acceptable to submit parts verbatim. Such use must be explicitly acknowledged in a README.
- In every homework, your README file must note what work is new, what work is based on work from a prior semester, and what work is submitted verbatim from a prior semester. Even if nothing is from a prior semester, your README file must disclose this information.
- In every homework, you must cite collaboration with every partner with whom you worked on the assignment in a past semester—even if none of that partner work survives.

Academic Honesty

Any violation of the above policies will be considered cheating, and all students suspected of being involved in any capacity will be forwarded directly to the Office of Student Affairs, who will investigate the case independently. Their sanctions range from horrible to inconceivably horrible. It's not worth it. We do use automated tools to detect plagiarism as well.

Inclusivity, Accessibility, and Additional Help

This course strives for inclusion of all participants, regardless of personal identity (gender, race, sexual orientation, religion, etc.), socio-economic background, disabilities, or neurodivergence. In the classroom and our discussion forums, everyone is expected to treat everyone else with dignity and respect. If you feel unwelcome or mistreated for any reason by either another student, a TA, or the course itself, please let an appropriate member of the teaching staff know so we can work to make things better.

CS 105 can be especially difficult for first-generation college students and for members of historically underrepresented groups in computing, who may not have the family or social support that helps them develop their skills in "how to be a college student." If you are a student in either of these categories, you are strongly encouraged to meet with a course instructor or contact one early in the term to talk about your support system.

Accommodations for Students with Disabilities

Tufts University values the diversity of our body of students, staff, and faculty and recognizes the important contribution each student makes to our unique community. Tufts is committed to providing equal access and support to all qualified students through the provision of reasonable accommodations so that each student may fully participate in the Tufts experience. If you have a disability that requires reasonable accommodations, please contact the StAAR Center at StaarCenter@tufts.edu or head to the StAAR Center website to make an appointment with an accessibility representative to determine appropriate accommodations. **Please be aware that accommodations cannot be enacted retroactively; all accomodation notes must be provided to the instructor within one week of the note being written to guarantee consideration.**

Academic Support at the StAAR Center

The StAAR Center offers a variety of resources to all students (both undergraduate and graduate) in the Schools of Arts and Sciences, and Engineering, the SMFA, and The Fletcher School; services are free to all enrolled students. Students may make an appointment to work on any writing-related project or assignment, attend subject tutoring in a variety of disciplines, or meet with an academic coach to hone fundamental academic skills like time management or overcoming procrastination. Students can make an appointment for any of these services by visiting tutorfinder.studentservices.tufts.edu, or by visiting students.tufts.edu/staar-center.

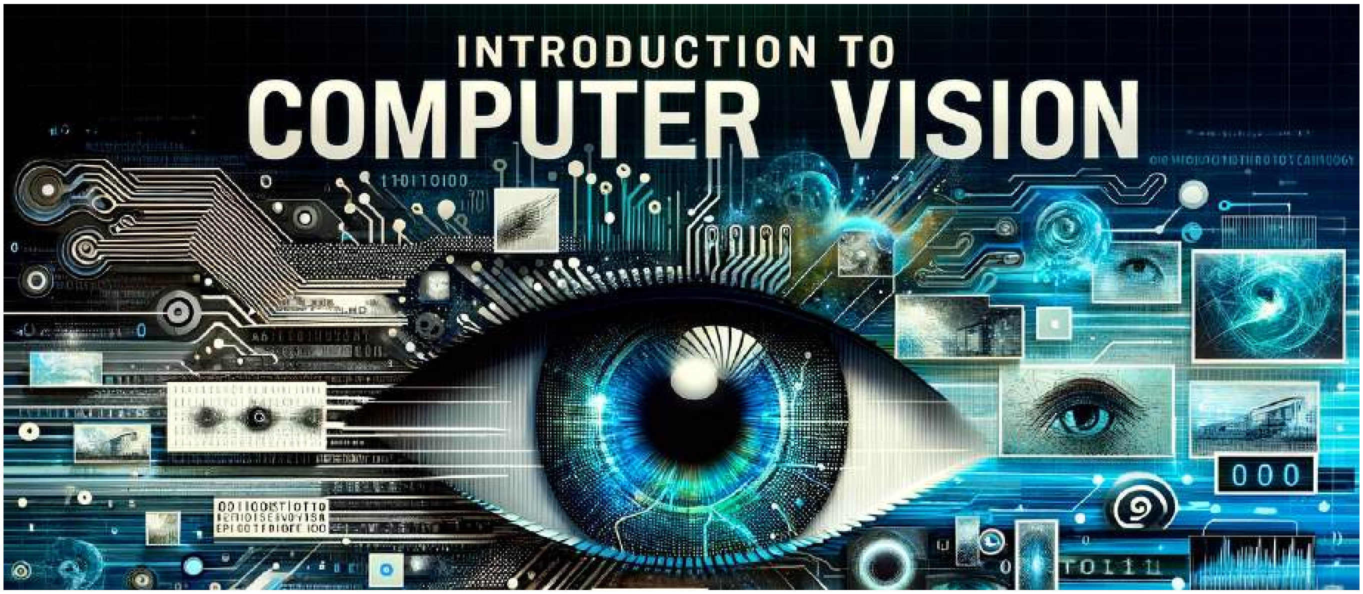
Important note on exam accommodations: All students approved for extended-time and/or distraction-reduced environment on their exams must book their exam space at the StAAR Center, no later than 10 days before the exam date. Students who do not comply with these guidelines and timeframes set by the StAAR Center may have to take their exam without any accommodation.

Religious Holy Days

We will reasonably accommodate any student who, for reasons of observing religious holy days, will be absent from recitation or experience any hardship in the completion of their work during the holy days. Please contact the instructor in advance of the holy day if you will miss recitation or if you need any additional accommodation (such as for assignments); reasonable accommodations will be allowed at no penalty **only if the instructor is notified well in advance**.

Acknowledgments

This syllabus (and the entire course) uses many ideas and materials developed by previous instructors of CS 105, including Norman Ramsey, Kathleen Fisher, Jeff Foster, and Richard Townsend. Thank you!



Introduction to Computer Vision

CS152-02 DS-153 | Spring 2024

This course is an introduction to low and intermediate level of classic and modern Computer Vision. We will learn how to design algorithms that process visual scenes to automatically extract information. The course will cover fundamental principles and important applications of computer vision, including image formation, processing, detection and matching features, image segmentation, and multiple views. We will also cover the basics of machine learning and deep learning for computer vision. The course will include a mix of lectures, paper readings, and hands-on Python programming assignments.

Cover photo hallucinted by ChatGPT

Course Information

Hours: Monday & Wednesday, 4:30PM – 5:45PM

Location: Joyce Cummings Center, room 160

Instructor: Dr. Roy Shilkrot roy.shilkrot@tufts.edu | roy@cs.tufts.edu | LinkedIn | GitHub | Twitter/X | YouTube

Office hours: Schedule online with Prof. Roy: https://calendly.com/royshilkrot/15min

Prerequisites: Algorithms and data structures, linear algebra (I or II), statistics and probability theory (I), calculus (I), working knowledge of Python

Teaching assistant:

- Yukun Li Yukun.Li@tufts.edu
- Fox Huston fox.huston@tufts.edu

TA Office hours: TBD

Piazza: https://piazza.com/tufts/spring2024/sp24cs015202ds0153/home

Canvas: https://canvas.tufts.edu/courses/54844

Course Goals

By the end of the semester, students should be able to:

- Identify and understand the fundamental aspects and techniques of Computer Vision
- Interpret and analyze a given problem using these techniques
- Adopt and Implement the appropriate techniques to solve a given problem

Textbooks

We will follow chapters from the books below:

- Computer Vision: Algorithms and applications, 2nd Ed (2022). Richard Szeliski. The book PDF is freely available online (http://szeliski.org/Book)
- Simon J.D Prince. Computer Vision Models (2012)
- Aeorlien Geron. Hands on Machine Learning 3rd ed. (2022)
- Francois Chollet. Deep Learning with Python 2nd ed. (2021)
- Excerpts from Hartley & Zisserman (2004), and Marr (2010)

As well as reference videos from Prof. Shree Nayar's lecture series First Principles of Computer Vision (https://www.youtube.com/@firstprinciplesofcomputerv3258)

Syllabus

Class	Content	Links	Assignment
1	Introduction: Class, Computer Vision, Math	slides video	
2	Image Formation: Human Vision, Optics, Physics, Cameras	slides video	
3	Image Formation cont.: Perspective projection, Pinhole camera model, Digital Imaging, Colorspaces	slides video	HW1: Hello Vision World
4	Image Processing: Sampling, Filters, Edges: Convolutions, Separability, Local operators, Frequency domain and Fourier Transform, Histograms, non-linear (median, bilateral), Pyramids	slides video	
5	Features: Interests points, Corners, Harris, Blobs, FAST, Multiscale	slides video	HW2: Pyramid blending, Corners & Blobs, Key points
6	Features: Descriptors, Gradient Histograms, Edges & Contours, Binary Images	slides video	

Class	Content	Links	Assignment
7	Geometric / Planar Transforms Parameterization, Feature Matching, Least Squares optimization	slides	HW3: Feature Extraction & Matching, Binary Images
8	Image Alignment, RANSAC, Stitching, Panoramas, Cylindrical coordinates, Global optimization / Bundle Adjustment, Blending	slides video	
9	Tracking: Meanshift, Motion estimation parameterizations	slides video	HW4: Panorama Stitching
10	Tracking: Correlation Filters, Optical flow, Lucas-Kanade, Shi-Tomasi	slides video	
11	Multiview 1: Intro, Camera Calibration, Calibration and Projection Matrices	slides video	
12	Multiview 2: Parallel Motion Stereo, Depth and Disparity, Epipolar Geometry, Essential and Fundamental Matrices	slides video	
13	Multiview 3: Triangulation, Camera Pose Estimation, Augmented Reality	slides video	HW5: Tracking
14	Mid-term (in class)		
15	Bundle Adjustment, Intro to Visual Machine Learning	slides video	
16	Linear Models, Linear Regression and Classification, Logistic Regression	slides video	HW6: MVG, Bundle Adjustment
17	Linear SVMs, Neural Networks, Gradient descent, Backpropagation and Learning Practices	slides video	
18	Convolutional Neural Networks, Deep Learning	slides video	HW7: Hello CNNs, Classification
19	Object detection 1: V-J, Intro to CNN object detection	slides video	
20	Object detection 2: Region Proposals Nets, R-CNN, SPP, Fast-RCNN, YOLO, SSD	slides video	
21	Segmentation I: Clustering, Flood-fills, Superpixels, Graph-cuts	slides video	HW8: Object Detection
22	Segmentation II: Semantic, FCNs, Losses, U-Net, Transpose Conv	slides video	
23	Vision and Language: Captioning, VQA, CNN+RNN	slides video	
24	Pose Estimation I: ASM, AAM, DPM, 2.5D Pose	slides video	HW9: Segmentation and Pose
25	Pose Estimation II: DeepPose, Heatmaps, PAFs, HRNet	slides video	

Class	Content	Links	Assignment
26	Final class. What have we missed? Further reading, final exam prep		
	Final exam: 5/8 from 12:00pm to 2:00pm, JCC 160		

Problem Sets, Tests

Problem sets: Eight (8) (or Nine (9)) problem sets will be given during the class. Any material regarding the set solution will be submitted electronically. Homework will be due up to one or two weeks from the assigned date at midnight. Late assignments are penalized at 20% for each 24 hours' delay. No homework will be accepted one week after the deadline. If a serious illness or another major life event prevents you from completing a homework assignment on time, you should report the event to your instruction team, after which alternate arrangements can be made. Reporting must be done before the assignment in question is due.

Tests: There will be two (2) tests over the course of the semester. The tests will be a combination of multiple choices and open-ended answers. If, for any serious illness or major life event you must miss a test, you must inform me before the day of the exam so we can work out an arrangement. Please Note: If you wish to dispute a grade, it is mandatory that you do so within one week of receiving the grade. After such a term, the grade will be considered final. Note: Grading is generous, re-grading is strict.

Final grades: You must show proficiency in all grading areas to pass the class. A failing average (below 50%) in any of the grading areas (problem sets, tests, attendance) will result in a failing grade in the class. Your final grade will be determined using the following percentage breakdown:

Grading Area	Percentage
Problem Sets	50%
Tests	40% (mid 13%, final 27%)
Attendance, Behavior, Attitude, Decorum	10%

Communications

Piazza will be our primary means of communication. All course announcements will also be made through Piazza. Rather than emailing questions to the teaching staff, we encourage you to post your questions as a public discussion. You are also encouraged to help each other, as long as the question does not contain any code or portion of a problem set answer. In such a case, the question must be made private (please, refer to the section Academic Honesty below). To schedule office hours outside the posted times, please email any of the teaching staff or post a private message on Piazza.

Academic Honesty

Science is, to its core, a collaborative effort. The advantages of coming together for examination or comparison, sometimes even just to explain the problem, are well known. I strongly encourage students to discuss course material, problems, and applications outside the classroom with the teaching staff and other students. You are also encouraged to form study groups for the tests.

Having said that, integrity and honesty are equally important qualities of any future academic, scientist or engineer. We take plagiarism very seriously. You must do homework, the problem sets and the final projects on your own. If you need help, the teaching staff will be more than happy to help you!

Using AI. We encourage and expect you to use AI tools such as GitHub Copilot and ChatGPT to aid you in this class, as would be expected of you by your future employers or colleagues. However, understand the risks of using these tools without a solid understanding of their inner workings. Research hallucinations and AI plagiarism before using any products of AI tools as submissions in this class. OpenAI has helpful information on the topic, as well as the faculty in Wharton.

Academic Integrity Policy

Tufts holds its students strictly accountable for adherence to academic integrity. The consequences for violations can be severe. It is critical that you understand the requirements of ethical behavior and academic work as described in Tufts' Academic Integrity handbook. If you ever have a question about the expectations concerning a particular assignment or project in this course, be sure to ask me for clarification. The Faculty of the School of Arts and Sciences and the School of Engineering are required to report suspected cases of academic integrity violations to the Dean of Student Affairs Office. If I suspect that you have cheated or plagiarized in this class, I must report the situation to the dean.