

**ECE 750T39**  
**Scalable Computer System Design**  
**Fall 2024**

Instructor: Dr. Tahsin Reza (tahsin.reza@uwaterloo.ca)

**Calendar Description.** In Computer Systems, the “scalability” property of a software solution may refer to: (i) its ability to improve performance (i.e., time-to-solution or throughput) by harnessing additional computing resources, e.g., multiple CPUs on a server or nodes of a compute cluster; (ii) the system’s capacity in maintaining a linear trend in performance with respect to growing problem size (i.e., input data or algorithm state); (iii) for application services, the ability to guarantee “availability” and “quality-of-service” with increasing number of clients or service requests. Scalable software solutions are crucial in the age of big data - modern problems like knowledge discovery and predictive analysis often involve tens of terabytes to petabytes of data. The importance of scalability is paramount in the cloud domain: today, there exist hundreds of cloud services with massive userbases - as large as two billion users. This course focuses on the, immensely important, “scalability” aspects of large-scale (parallel/distributed) software design; it is organized to help the audience, (i) comprehend scalability demands of large-scale problems; (ii) familiarize with state-of-the-art (scalable) systems approaches in key problem classes; (iii) acquaint with related computing platforms, and tools and their applications to solution design.

**Schedule**

Day and Time	Location
Tuesday, 8:30AM – 9:50AM	████████
Thursday, 8:30AM – 9:50AM	████████

**Course Topics** (topics covered in the course and delivery methods will be adjusted based on the size of the class)

1. Concepts of scalability in computer systems (3 hours)
  - What scalability means in different contexts?
  - Understanding key scalability metrics.
  - Analytical performance models concerning scalability, e.g., Amdahl’s/Gustafson’s law, LogP [1, 2].
2. Overview of scalable computing platforms and development tools (7 hours)
  - Scalable multi-processor architectures, e.g., SMP vs. NUMA.
  - Scalable network topology - enabler for scalable distributed computing, e.g., Fat tree, Torus, and Dragonfly [3, 4, 24, 25].
  - A quick introduction to parallel/distributed computation models (e.g., MIMD, SPMD, PGAS and Dataflow) and implementation/runtime frameworks (e.g., OpenMP, MPI, SHMEM and Hadoop/Spark) [5].
3. Fundamental (systems) design concepts (2 hours)
  - Workers, data partitioning, task distribution and load balancing; understanding common scalability bottlenecks and how to address them.
4. Study of scalable solution design for key problem classes, for example (7.5 hours)
  - Linear algebra: distributed Sparse Matrix Vector multiplication (SpMV) [10].
  - Numerical analysis: distributed Fast Fourier Transform (FFT) [11].
  - Sequence alignment: distributed De novo genome assembly [12].
  - Distributed processing of massive scale-free graphs: understanding scalability challenges of highly irregular problems; graph partitioning, load balancing techniques and distributed communication optimization [13, 14, 15].
5. Scalable storage solution design (4.5 hours)
  - Distributed file systems [6, 7].
  - Distributed data stores [8, 9].
6. Cloud computing (4.5 hours)
  - Geo-distributed caching - a key enabler for scalable cloud services [8, 16, 17].
  - Serverless computing - a modern approach to scalable cloud services [18, 19, 20].

## 7. Scalable deep learning design approaches (4.5hours)

- Data parallel vs. model parallel training [21]; centralized vs. decentralized model optimization [22]; synchronous vs. asynchronous parameter exchange in centralized and decentralized optimization approaches [23].

**Grading.** Four assignments 20%, two term-tests 30%, leading paper discussions 5%, project proposal 5%, project presentation and demo 20%, and project report 20%.

**Course Assignments** (number and type of deliverables will be adjusted based on the size of the class)

*A short programming assignment* (either individual or group, 10%). Composed of two deliverables: (i) Implementation of a parallel/distributed computing solution focusing on scalability - ideally on top of an existing middleware/high-level framework; (ii) Scalability evaluation following one of the performance models covered in the course.

*Writing paper reviews* (individual, 10%). Students will write reviews of three peer-reviewed research papers; maximum one page in length including summary and critical assessment of the work: students are required to discuss key contributions, potential gaps in the proposed solution and evaluation methodology, and suggest improvements.

**Course Project.** An elaborate group project that exercises the steps involved in designing, implementing and evaluating a scalable computing solution. Student groups have the flexibility of choosing the problem, hardware platform (parallel shared memory or shared-noting distributed) and implementation framework. There are three deliverables: (i) project proposal - a brief description of the problem of interest, its scalability requirements, overview of anticipated solution design and work plan; (ii) project presentation - a 20-minute group presentation of the completed project; (iii) project report - description of design choices, implementation approach, evaluation methodology and results, and future work. Depending on the class size, each group would have 2 to 5 members.

**Intended Learning Outcomes.** Upon successful completion of this course, students should have: (i) learned about the significance of scalability in computer systems; (ii) a grasp on key scalable computing concepts; (iii) familiarized with the design of scalable computing hardware platforms; (iv) learned about unique scalability requirements in key application areas and state-of-the-art solution techniques for these problem; (v) familiarized with key parallel/distributed programming models and implementation/runtime frameworks; (vi) acquired fundamental skills to reason about the design of a scalable software system and hands-on experience of system building; (vii) acquainted with resources for future learning and research.

**Required background and prerequisites.** It is expected that students have already completed undergraduate computer science/engineering courses that covered the following topics: data structures, algorithms, object-oriented programming, operating systems, computer architecture and computer networks. Although not required, having taken one of the following courses, ECE 454, ECE 459 or ECE 751, would certainly help, especially, with the programming assignment and term project.

**Learning Resources.** Slide decks used during lectures, research articles for paper review and other reading materials will be provided by the instructor. While no formal text book is required, students may find the following books useful:

- [Principles of Computer System Design: An Introduction](#), Jerome H. Saltzer, M. Frans Kaashoek, 1st edition
- [Distributed Systems: Principles and Paradigms](#), Andrew S. Tanenbaum, Maarten van Steen, 4th edition
- [Distributed Systems: Concepts and Design](#), George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, 5th edition

This course will use LEARN, as well as Piazza as a discussion forum. Important announcements will be posted on LEARN, including a signup link for Piazza. Please use Piazza exclusively to communicate with the instructor or the TAs regarding the course.

## References

[1] X.H. Sun and L.M. Ni, "[Scalable Problems and Memory-Bounded Speedup](#)", Journal of Parallel and Distributed Computing (JPDC), Volume 19, Issue 1, Elsevier B.V., 1993.

[2] Kenneth Czechowski and Richard Vuduc, "[A theoretical framework for algorithm-architecture co-design](#)", IEEE International Parallel and Distributed Processing Symp. (IPDPS'13), Boston, MA, 2013.

- [3] John Kim, William J. Dally, Steve Scott, and Dennis Abts, “[Technology-Driven, Highly-Scalable Dragonfly Topology](#)”, IEEE International Symposium on Computer Architecture (ISCA'08), Beijing, China, 2008.
- [4] Abhinav Bhatele, Nikhil Jain, Misbah Mubarak, and Todd Gamblin, “[Analyzing Cost-Performance Tradeoffs of HPC Network Designs under Different Constraints using Simulations](#)”, ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS'19), New York, NY, 2019.
- [5] John Bachan et al., “[UPC++: A high-performance communication framework for asynchronous computation](#)”, IEEE International Parallel and Distributed Processing Symp. (IPDPS'19), Rio de Janeiro, Brazil, 2019.
- [6] “[Lustre: A Scalable High-Performance File System](#)”, Whitepaper, Cluster File Systems Inc., 2003.
- [7] Satadru Pan et al., “[Facebook's Tectonic Filesystem: Efficiency from Exascale](#)”, USENIX Conference on File and Storage Technologies (FAST'21), virtual, 2021.
- [8] David Karger et al., “[Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web](#)”, ACM symposium on theory of computing (STOC'97), New York, NY, 1997.
- [9] Giuseppe DeCandia et al., “[Dynamo: amazon's highly available key-value store](#)”, ACM SIGOPS symposium on Operating systems principles (SOSP'07), New York, NY, 2007.
- [10] Han. D. Tran et al., “[A scalable adaptive-matrix SPMV for heterogeneous architectures](#)”, IEEE International Parallel and Distributed Processing Symposium (IPDPS'22), Lyon, France, 2022.
- [11] Dmitry Pekurovsky, “[P3DFFT: A Framework for Parallel Computations of Fourier Transforms in Three Dimensions](#)”, SIAM Journal on Scientific Computing 2012 34:4, C192-C209.
- [12] Evangelos Georganas et al., “[HipMer: an extreme-scale de novo genome assembler](#)”, IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis (SC'15), Austin, TX, 2015.
- [13] Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos, “[R-MAT: A Recursive Model for Graph Mining](#)”, SIAM International Conference on Data Mining (SDM'04), Lake Buena Vista, Florida, 2004.
- [14] Roger Pearce, Maya Gokhale, and Nancy M. Amato, “[Faster parallel traversal of scale free graphs at extreme scale with vertex delegates](#)”, IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis, (SC'14), New Orleans, LA, 2014.
- [15] Roshan Dathathri, Keshav Pingali, Marc Snir et al., “[Gluon-Async: A Bulk-Asynchronous System for Distributed and Heterogeneous Graph Analytics](#)”, International Conference on Parallel Architectures and Compilation Techniques (PACT'19), Seattle, WA, 2019.
- [16] Qi Huang et al., “[An analysis of Facebook photo caching](#)”, ACM Symposium on Operating Systems Principles (SOSP'13), New York, NY, 2013.
- [17] Behrouz Zolfaghari et al., “[Content Delivery Networks: State of the Art, Trends, and Future Roadmap](#)”, ACM Computing Surveys, 53, 2, Article 34 (March 2021), 34 pages.
- [18] Mohammad Shahradd, Rodrigo Fonseca, Íñigo Goiri, Gohar Chaudhry, Paul Batum, Jason Cooke, Eduardo Laureano, Colby Tresness, Mark Russinovich, and Ricardo Bianchini, “[Serverless in the wild: characterizing and optimizing the serverless workload at a large cloud provider](#)”, USENIX Annual Technical Conference (USENIX ATC'20), virtual, 2020.
- [19] Johann Schleier-Smith, Joseph E. Gonzalez, Ion Stoica, David A. Patterson et al., “[What serverless computing is and should become: the next phase of cloud computing](#)”, Communications of the ACM 64, 5 (May 2021), 76-84.
- [20] Samuel Kounev, Nikolas Herbst, Cristina L. Abad, Alexandru Iosup, Ian Foster, Prashant Shenoy, Omer Rana, and Andrew A. Chien, “[Serverless Computing: What It Is, and What It Is Not?](#)” Communications of the ACM 66, 9 (September 2023), 80-92.
- [21] Eric P. Xing et al., “[Petuum - A New Platform for Distributed Machine Learning on Big Data](#)”, IEEE Transactions on Big Data, vol. 1, no. 2, pp. 49-67, 2015.

- [22] Xiangru Lian et al., "[Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent](#)", International Conference on Neural Information Processing Systems (NeurIPS'17), Red Hook, NY, 2017.
- [23] Shuxin Zheng et al., "[Asynchronous Stochastic Gradient Descent with delay compensation](#)", International Conference on Machine Learning (ICML'17), Sydney, Australia, 2017.
- [24] Torsten Hoefler, Ariel Hendel, and Duncan Roweth, "[The Convergence of Hyperscale Data Center and High-Performance Computing Networks](#)", IEEE Computer, vol. 55, no. 7, pp. 29-37, 2022.
- [25] Alexander Shpiner, et al., "[Dragonfly+: Low Cost Topology for Scaling Datacenters](#)", IEEE Int. W/S on High-Perf. Interconnection Networks in the Exascale and Big-Data Era (HiPINEB'17), Austin, TX, 2017.

## Course Policy

**Lateness and absence:** Students who miss lectures will be responsible for all content covered and are encouraged to catch up by studying the provided lectures slides, and course textbook, as well as by consulting classmates. Late assignment submissions will normally be penalized 1% per hour. There are no grace days, but short-term absence declarations are applicable to assignments. Once the latter have been used up (one two-day absence per student per term), the instructor reserves the right to reconsider penalties for students who are struggling due to illness or personal issues. Please approach the instructor discretely if you wish to be considered for compassionate accommodation.

**Collaboration and group work:** Assignments will be completed in groups of up to 3 students. It is OK to work in a group of 1 (i.e., alone), but you will not receive any extra credit for the additional work you undertake. Group enrolment will be done separately for each assignment in LEARN. Lightweight discussion between individuals and assignment groups is permitted, but solutions should be developed independently.

Each group member, individually is responsible for learning all the material corresponding to a group deliverable, and may be required to answer technical questions posed by members of the teaching team. Members of the teaching team may assign or rearrange groups for academic reasons. Copying material from other groups, from other courses, or from online sources is forbidden except for materials authorized explicitly by the course instructor. Code should never be shared electronically with classmates except among members of the same assignment group, or with the instructor's permission. The source of any legitimately copied material must be acknowledged, for example by adding comments to source code. Unauthorized collaboration and copying of material constitute academic offences under [Policy 71](#). Your code may be verified using a plagiarism checker such as MOSS (Measure Of Software Similarity).

**Piazza:** Please be respectful and polite when posting on Piazza. Feel free to ask for clarification regarding the requirements of an assignment, or ask general technical questions, but do not post information (e.g., code fragments) that gives away a substantial part of the solution.

**Submission and pickup of assignments:** Assignment deliverables will be submitted electronically using a dropbox in LEARN. For group assignments, **you must enroll in a group before the dropbox becomes accessible**. Feedback from graders may be provided on paper, electronically either by email, or using the same dropbox in LEARN.

**Grading and regrading:** All coursework items, including exams, will be graded on the basis of completeness, correctness, and clarity. Ask questions if you need clarifications.

Where appropriate, students are expected to test their own solutions to problems using provided or instructed inputs. If a student's solution differs substantially from the solution(s) accepted by the teaching team the student may be asked to provide sample inputs and outputs to evaluate the correctness of their solution.

Students who feel that they have been graded unfairly may request that a coursework item be regraded, in which case the entire item (i.e., assignment or exam) will be regraded. As a result, the grade may increase, decrease, or remain unchanged. The teaching team may refuse to regrade coursework if there is substantial concern that the solution was modified or otherwise tampered with after submission. Alternatively, the teaching team may accept a modified solution for regrading and apply a penalty.

## Assignment Screening

Copying material from other groups, from other courses, or from online sources is forbidden except for materials authorized explicitly by the course instructor. Code should never be shared electronically with classmates except among members of the same assignment group, or with the instructor's permission. The source of any legitimately copied material must be acknowledged, for example by adding comments to source code. Unauthorized collaboration and copying of material constitute academic offences under Policy 71. Your code may be verified using a plagiarism checker such as MOSS (Measure Of Software Similarity).

## Administrative Policy

**Illness:** Please self-isolate as needed when experiencing a contagious illness. The instructor may, at their discretion, livestream lectures for students who are unable to attend in person. Additional accommodations such as due date extensions and re-weighting of deliverables will be considered in accordance with existing department-level, faculty-level, and university-level policies. Missed deliverables due to prolonged illness will normally be replaced with the final exam grade. Students with chronic health problems that affect their ability to complete coursework should register with [AccessAbility Services](#).

**Fair contingencies for unforeseen circumstances:** This course outline presents the instructor's intentions for course assessments, their weights, and due dates in Spring 2024. As best as possible, we will keep to the specified assessments, weights, and dates. To provide contingency for unforeseen circumstances, the instructor reserves the right to modify course topics and/or assessments and/or weight and/or deadlines with due and fair notice to students. In the event of unexpected challenges, the instructor will work with the ECE Department and Faculty of Engineering to find reasonable and fair solutions that respect rights and workloads of students, staff, and faculty.

**Intellectual Property:** Learning materials used in this course contain the intellectual property of current and past members of the teaching team. Examples of such materials include lecture notes, lecture slides, sample exams and solutions, assignment specifications, and starter code. Much of this content is protected by copyright, and should not be distributed to websites (e.g., CourseHero) or online repositories (e.g., GitHub), or made publicly available in any other manner, without consent of the copyright holder(s). It is reasonable, however, to create copies of the copyrighted materials on your personal computing device for your own use, and to copy starter code to your home directory in [ecelinux](#).

**Generative AI:** Generative artificial intelligence (GenAI) trained using large language models (LLM) or other methods to produce text, images, music, or code, like Chat GPT, DALL-E, or GitHub CoPilot, may be used in this course with proper documentation, citation, and acknowledgement. Permitted uses of and expectations for using GenAI will be discussed in class and outlined on assignment instructions.

Recommendations for how to cite generative AI in student work at the University of Waterloo may be found through the Library: [https://subjectguides.uwaterloo.ca/chatgpt\\_generative\\_ai](https://subjectguides.uwaterloo.ca/chatgpt_generative_ai). Please be aware that generative AI is known to falsify references to other work and may fabricate facts and inaccurately express ideas. GenAI generates content based on the input of other human authors and may therefore contain inaccuracies or reflect biases.

In addition, you should be aware that the legal/copyright status of generative AI inputs and outputs is unclear. Exercise caution when using large portions of content from AI sources, especially images. More information is available from the Copyright Advisory Committee: <https://uwaterloo.ca/copyright-at-waterloo/teaching/generative-artificial-intelligence>

You are accountable for the content and accuracy of all work you submit in this class, including any supported by generative AI.

## Faculty of Engineering Guiding Practices

**Territorial Acknowledgement:** The University of Waterloo acknowledges that much of our work takes place on the traditional territory of the Neutral, Anishinaabeg and Haudenosaunee peoples. Our main campus is situated on the Haldimand Tract, the land granted to the Six Nations that includes six miles on each side of the Grand River. Our active

work toward reconciliation takes place across our campuses through research, learning, teaching, and community building, and is centralized within the [Office of Indigenous Relations](#).

**Inclusive Teaching-Learning Spaces:** The University of Waterloo values the diverse and intersectional identities of its students, faculty, and staff. The University regards equity and diversity as an integral part of academic excellence and is committed to accessibility for all. We consider our classrooms, online learning, and community spaces to be places where we all will be treated with respect, dignity, and consideration. We welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability – and other visible and nonvisible differences. We are all expected to contribute to a respectful, welcoming, and inclusive teaching-learning environment. Any member of the campus community who has experienced discrimination at the University is encouraged to seek guidance from the [Office of Equity, Diversity, Inclusion & Anti-racism \(EDI-R\)](#) via email at [equity@uwaterloo.ca](mailto:equity@uwaterloo.ca). [Sexual Violence Prevention & Response Office \(SVPRO\)](#), supports students at UWaterloo who have experienced, or have been impacted by, sexual violence and gender-based violence. This includes those who experienced harm, those who are supporting others who experienced harm. SVPRO can be contacted at [svpro@uwaterloo.ca](mailto:svpro@uwaterloo.ca)

**Religious & Spiritual Observances:** The University of Waterloo has a duty to accommodate religious and spiritual observances under the Ontario Human Rights Code. Please inform the instructor at the beginning of term if special accommodation needs to be made for religious observances that are not otherwise accounted for in the scheduling of classes and assignments. Consult with your instructor(s) within two weeks of the announcement of the due date for which accommodation is being sought.

**Respectful Communication and Pronouns:** Communications with Instructor(s) and teaching assistants (TAs) should be through recommended channels for the course (e.g., email, LEARN, Piazza, Teams, etc.) Please use your UWaterloo email address. Include an academic signature with your full name, program, student ID. We encourage you to include your pronouns to facilitate respectful communication (e.g., he/him; she/her; they/them). You can update your chosen/preferred name at [WatIAM](#). You can update your pronouns in [Quest](#).

**Mental Health and Wellbeing Resources:** If you are facing challenges impacting one or more courses, contact your academic advisor, Associate Chair Undergraduate, or the Director of your academic program. Mental health is a serious issue for everyone and can affect your ability to do your best work. We encourage you to seek out mental health and wellbeing support when needed. The [Faculty of Engineering Wellness Program](#) has programming and resources for undergraduate students. For counselling (individual or group) reach out to [Campus Wellness and Counselling Services](#). Counselling Services is an inclusive, non-judgmental, and confidential space for anyone to seek support. They offer confidential counselling for a variety of areas including anxiety, stress management, depression, grief, substance use, sexuality, relationship issues, and much more.

**Intellectual Property:** Be aware that this course contains the intellectual property of their instructor, TA, and/or the University of Waterloo. Intellectual property includes items such as:

- Lecture content, spoken and written (and any audio/video recording thereof).
- Lecture handouts, presentations, and other materials prepared for the course (e.g., PowerPoint slides).
- Questions or solution sets from various types of assessments (e.g., assignments, quizzes, tests, final exams); and
- Work protected by copyright (e.g., any work authored by the instructor or TA or used by the instructor or TA with permission of the copyright owner).

Course materials and the intellectual property contained therein are used to enhance a student's educational experience. However, sharing this intellectual property without the intellectual property owner's permission is a violation of intellectual property rights. For this reason, it is necessary to ask the

instructor, TA and/or the University of Waterloo for permission before uploading and sharing the intellectual property of others online (e.g., to an online repository).

Permission from an instructor, TA or the University is also necessary before sharing the intellectual property of others from completed courses with students taking the same/similar courses in subsequent terms/years. In many cases,

instructors might be happy to allow distribution of certain materials. However, doing so without expressed permission is considered a violation of intellectual property rights and academic integrity.

Please alert the instructor if you become aware of intellectual property belonging to others (past or present) circulating, either through the student body or online.

**Continuity Plan - Fair Contingencies for Unforeseen Circumstances (e.g., resurgence of COVID-19):** In the event of emergencies or highly unusual circumstances, the instructor will collaborate with the Department/Faculty to find reasonable and fair solutions that respect rights and workloads of students, staff, and faculty. This may include modifying content delivery, course topics and/or assessments and/or weight and/or deadlines with due and fair notice to students. Substantial changes after the first week of classes require the approval of the Associate Dean, Undergraduate Studies.

**Declaring absences:** *[undergraduate students and/or courses only]* Regardless of the process used to declare an absence, students are responsible for reaching out to their instructors as soon as possible. The course instructor will determine how missed course components are accommodated. Self-declared absences (for COVID-19 and short-term absences up to 2 days) must be submitted through [Quest](#). Absences requiring documentation (e.g., Verification of Illness Form, bereavement, etc.) are to be uploaded by completing the form on the [VIF System](#). The [UWaterloo Verification of Illness form](#), completed by a health professional, is the only acceptable documentation for an absence due to illness. Do not send documentation to your advisor, course instructor, teaching assistant, or lab coordinator. Submission through the VIF System, once approved, will notify your instructors of your absence.

**Rescheduling Co-op Interviews:** Follow the co-op process for [rescheduling co-op interviews](#) for conflicts to graded assignments (e.g., midterms, tests, and final exams). Attendance at co-operative work-term employment interviews is not considered to be a valid reason to miss a test.

## University Policy

**Academic integrity:** In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. [Check [the Office of Academic Integrity](#) for more information.]

**Grievance:** A student who believes that a decision affecting some aspect of their university life has been unfair or unreasonable may have grounds for initiating a grievance. Read [Policy 70, Student Petitions and Grievances, Section 4](#). When in doubt, please be certain to contact the department's administrative assistant who will provide further assistance.

**Discipline:** A student is expected to know what constitutes academic integrity to avoid committing an academic offence, and to take responsibility for their actions. [Check [the Office of Academic Integrity](#) for more information.] A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate associate dean. For information on categories of offences and types of penalties, students should refer to [Policy 71, Student Discipline](#). For typical penalties, check [Guidelines for the Assessment of Penalties](#).

**Appeals:** A decision made or penalty imposed under [Policy 70, Student Petitions and Grievances](#) (other than a petition) or [Policy 71, Student Discipline](#) may be appealed if there is a ground. A student who believes they have a ground for an appeal should refer to [Policy 72, Student Appeals](#).

**Note for students with disabilities:** [AccessAbility Services](#), located in Needles Hall, Room 1401, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with AccessAbility Services at the beginning of each academic term.

**Turnitin.com:** Text matching software (Turnitin®) may be used to screen assignments in this course. Turnitin® is used to verify that all materials and sources in assignments are documented. Students' submissions are stored on a U.S. server, therefore students must be given an alternative (e.g., scaffolded assignment or annotated bibliography), if they are

concerned about their privacy and/or security. Students will be given due notice, in the first week of the term and/or at the time assignment details are provided, about arrangements and alternatives for the use of Turnitin in this course.

It is the responsibility of the student to notify the instructor if they, in the first week of term or at the time assignment details are provided, wish to submit alternate assignment.