CSE 312: Foundations of Computing II

Winter 2022

Instructor:	Alex Tsun	**Time:**	MWF 9:30-10:20am PST
Email:	alextsun@cs.washington.edu	**Location:**	ARC 147

Course Resources:

1. Course Website: `https://courses.cs.washington.edu/courses/cse312/22wi`
2. Gradescope, Edstem, Calendar, and other materials linked from the website above.

Announcement: You should regularly check the class web site for announcements and other information, including the most up-to-date information on problem sets and errata. The class web page will also have the schedule of topics to be covered and links to other class materials, including accompanying lecture notes, slides (after lecture) etc. If you have any personal questions, please email me directly. Any other non-sensitive questions about course content should be posted on our discussion forum.

TAs and Office Hours: See website.

Textbooks:

- (Recommended) Alex Tsun, *Probability & Statistics with Applications to Computing*, 2020. Available free online here.
- (Optional) Sheldon Ross, *A First Course in Probability* (10th Ed.), Pearson Prentice Hall, 2018.
- (Optional) Larsen and Marx, *An Introduction to Mathematical Statistics* (5th edition). Prentice-Hall.
- (Optional) Dimitri P. Bertsekas and John N. Tsitsiklis, *Introduction to Probability*, First Edition, Athena Scientific, 2000. Available free online here.

Prerequisites: CSE 311 and MATH 126. Here is a quick rundown of some of the mathematical tools we'll be using in this class: calculus (integration and differentiation), linear algebra (basic operations on vectors and matrices), an understanding of the basics of set theory (subsets, complements, unions, intersections, cardinality, etc.), and familiarity with basic proof techniques (including induction).

Why is CSE 312 Important?

While the initial foundations of computer science began in the world of discrete mathematics (after all, modern computers are digital in nature), recent years have seen a surge in the use of probability as a tool for the analysis and development of new algorithms and systems. As a result, it is becoming increasingly important for budding computer scientists to understand probability theory, both to provide new perspectives on existing ideas and to help further advance the field in new ways.

Probability is used in a number of contexts, including analyzing the likelihood that various events will happen, better understanding the performance of algorithms (which are increasingly making use of randomness), or modeling the behavior of systems that exist in asynchronous environments ruled by uncertainty (such as requests being made to a web server). Probability provides a rich set of tools for modeling such phenomena and allowing for precise mathematical statements to be made about the performance of an algorithm or a system in such situations.

Furthermore, computers are increasingly often being used as data analysis tools to glean insights from the enormous amounts of data being gathered in a variety of fields; you've no doubt heard the phrase "big data" referring to this phenomenon. Probability theory and statistics are the foundational methods used for designing new algorithms to model such data, allowing, for example, a computer to make predictions about new or uncertain events. In fact, many of you have already been the users of such techniques. For example, most email systems now employ automated spam detection and filtering. Methods for being able to automatically infer whether or not an email message is spam are frequently rooted in probabilistic methods. Similarly, if you have ever seen online product recommendation (e.g., "customers who bought X are also likely to buy Y"), you've seen yet another application of probability in computer science. Even more subtly, answering detailed questions like how many buckets you should have in your a hash table or how many machines you should deploy in a data center (server farm) for an online application make use of probabilistic techniques to give precise formulations based on testable assumptions.

Our goal in this course is to build foundational skills and give you experience in the following areas:

1. **Understanding the combinatorial nature of problems**: Many real problems are based on understanding the multitude of possible outcomes that may occur, and determining which of those outcomes satisfy some criteria we care about. Such an understanding is important both for determining how likely an outcome is, but also for understanding what factors may affect the outcome (and which of those may be in our control).

2. **Working knowledge of probability theory and some of the key results in statistics**: Having a solid knowledge of probability theory and statistics is essential for computer scientists today. Such knowledge includes theoretical fundamentals as well as an appreciation for how that theory can be successfully applied in practice. We hope to impart both these concepts in this class.

3. **Appreciation for probabilistic statements**: In the world around us, probabilistic statements are often made, but are easily misunderstood. For example, when a candidate in an election is said to have a 53% likelihood of winning does this mean that the candidate is likely to get 53% of the vote, or that that if 100 elections were held today, the candidate would win 53% of them? Understanding the difference between these statements requires an understanding of the model in the underlying probabilistic analysis.

4. **Applications in machine learning and theoretical computer science**: We are not studying probability theory simply for the joy of drawing summation symbols (okay, maybe some people are, but that's not what we're really targeting in this class), but rather because there are a wide variety of applications where probability and statistics allow us to solve problems that might otherwise be out of reach (or would be solved more poorly without the tools that probability and statistics can bring to bear). We'll look at examples of such applications throughout the class. For example, machine learning is a quickly growing subfield of artificial intelligence which has grown to impact many applications in computing. It focuses on analyzing large quantities of data to build models that can then be harnessed in real problems, such as filtering email, improving web search, understanding computer system performance, predicting financial markets, or analyzing DNA. Probability and statistics form the foundation of these systems. Another example application area is the use of randomized algorithms and probabilistic data structures. These usually have simpler and more elegant implementations than their deterministic counterparts, and have more efficient time and/or space complexity. We will be learning about some of these applications and you will have the opportunity to implement some of these algorithms.

Mask policy: As we look forward to being back in person, it is important that we take every precaution to keep our community safe. To this end, note that all students, TAs and the professor are required to wear masks in class and in in-person office hours. This will be strictly enforced. Here is more information on the UW Face Covering Policy.

Goals for Winter 2022:
We fully recognize that this quarter will be strange in many ways – our second quarter back in person since early in the pandemic, while COVID-19 is still with us. We hope that being back in person will make this a better experience for you. On the other hand, the mask requirement might make lectures or sections harder to understand on occasion. Don't be shy about asking the professor or TAs to repeat something they said if you're having trouble making it out.

As always, we are determined to reach the following course goals to the best of our ability: (1) To maintain the intellectual rigor of the CSE 312 curriculum while providing flexible ways for you to learn, and (2) To foster and maintain human connections and a sense of community throughout this course.

Please bear in mind that none of us have fully adjusted to the new normal, and we may have to adapt throughout the quarter. Everyone needs support and understanding in this unprecedented time and we are here to listen to you. Please don't hesitate to let us know about any issues that arise. Thanks and welcome to CSE 312. (Credit for this wording goes to Brandon Bayne from UNC - Chapel Hill.)

Tentative Course Outline:

1. Combinatorial Theory
2. Discrete Probability
3. Discrete Random Variables
4. Continuous Random Variables
5. Multiple Random Variables
6. Concentration Inequalities
7. Statistical Estimation
8. Statistical Inference
9. Applications to Computing

Lectures:
We will be holding live lectures in ARC 147 on Mondays, Wednesdays, and Fridays, 9:30AM – 10:20AM Pacific Time. The lectures will be recorded on Panopto, and you will be able to access those recordings within a few hours after class on Canvas.

Grading Breakdown:

Problem Sets (5)	50%
Concept Checks	10%
Quizzes (3)	40%

Problem Sets

- There will be 5 problem sets, equally weighted. There will be a written part and a coding part involved in each.
- The written parts **must be typed using LaTeX**, and submitted to Gradescope. A tutorial will be provided and recorded by the end of the first week. If you take other classes that involve a fair amount of math (such as the machine learning class CSE 446) or plan to write research papers, you will need to typeset in LaTeX anyway. It is a very useful skill, so you may as well start now.

- You **must** show your work; at a minimum of 1-2 sentences per question, but ideally as much as you would need to explain to a fellow classmate who hadn't solved the problem before. Be concise. **A correct answer with no work is worth nothing, less than a wrong answer with some work.**
- You **must** tag the question parts of your homework correctly on Gradescope. Failure to do so will **result in a 0** on **every** untagged question. Please check your submission by clicking each question, and making sure your solution appears there. Tagging can be included or changed after the assignment is submitted and after it is due as well.
- The coding parts will be written in Python3, with no exceptions. This is because the coding parts will be autograded. There are no hidden tests, and you'll have unlimited attempts. Whatever you see last on Gradescope for that section will be your grade. The implementation you do will provide you with a deeper understanding of how the theory we learn in this class is used in practice and should be a lot of fun. Note that we do not expect you to have any experience or knowledge of Python – we will provide you with tutorials and other kinds of help to get you started. A huge bonus of this class will be that you will come away with basic, working knowledge of Python (which you will undoubtedly use in the future, and definitely if you take CSE 446, the machine learning class).
- Regrade requests are due on Gradescope within **one week** of grades being published.
- It is okay to collaborate in coming up with solutions, but you must list all your collaborators at the top of each homework.
- For some particularly challenging PSets, we will allow groups of up to 2. The exact PSets for group and individual work are yet to be determined. You cannot have the same partner more than once. We will occasionally offer social sessions and surveys for finding partners.
- **(Edstem) Discussion Board Etiquette**
  1. A single thread for each question in each PSet will be created by the staff (e.g., PSet2 Q5). All clarification questions which are not personal should be posted in the corresponding thread to avoid duplicate questions.
  2. All **new** threads will require a staff member's approval (but replying on a thread like mentioned above will not). If you have a specific/personal question (e.g., sharing a large part of the solution or your thought process), please make a private post.
- **Late Policy**
  1. Problem Sets 1-4: On-time problem sets will be given 2% extra credit. Each problem set will be accepted up to two days late without penalty, with a 25% deduction each day after. (After 6 days, problem sets will not be accepted as we will publish grades and solutions according to a fixed schedule.)
  2. Problem Set 5: Will **not** be accepted late, nor will it have an on-time bonus.

Concept Checks

- Immediately after each lecture, concept checks for the sections covered will appear on Gradescope (typically 1-2 per lecture). Concept checks are due **30 minutes before the next lecture.**
- You can submit your answers as many times as you want; we will only grade the final submission. Correct answers will reveal the answer explanation; all other answers will not, so you can keep trying until you see the answer explanation.
- Concept checks can be submitted up to one week late for up to half-credit.

Quizzes

- There will be 3 equally-weighted quizzes given in the quarter; dates will be posted on the course website, but they will occur approximately evenly spaced out, with the last quiz being administered during finals week. These will be administered remotely through Gradescope or a similar tool, not during class time.
- These will have a time limit of 2 hours, but are to be completed anytime contiguously in a certain 24-hour window.
- These are to be done individually, and will consist of some T/F, multiple choice, and short answers.
- Late quizzes will not be accepted, and will be recorded with a score of 0.

Attendance Policy: Regular attendance to lecture and section is strongly recommended. Moreover, I encourage you in the strongest possible terms to ask questions during lecture and sections (as well as in office hours and on the discussion board). That will make the class more fun for all and you will definitely learn more and have an easier time with the homework!! One way to ask questions during class is to post a question on Edstem: there will be a thread for each lecture, and 1-2 TAs will be monitoring it then. They will either answer your question on Edstem or interrupt Alex to have him address the question.

Keep in mind that this class is fast-paced, and the problem sets will be challenging. To incentivize attendance, some problem set questions will be explicitly solved **during lectures or sections**. Furthermore, sections will be typically devoted to either heavily guiding you in the coding portions of the psets or giving you practice on problems similar to those on the PSets.

That being said, lectures will always be recorded. If you are feeling sick or experiencing symptoms, please do not come to lecture or section. We will do our best to provide any resources we can to help minimize the impact.

Academic Integrity: Lack of knowledge of the academic honesty policy is not a reasonable explanation for a violation. Each student is expected to do their own work on the problem sets in CSE 312. Students may discuss problem sets with each other as well as the course staff, with the following caveats:

- Do not take away any notes or screenshots from your discussions with others.
- After discussing with others, take a 30 minute break before writing up your solutions.
- Cite the names of all your collaborators at the top on your problem set.
- On a solo problem set, write up your solutions entirely on your own. On a problem set done in pairs, the pair should be writing up their solution together, but without any consultation with other pairs.

Excessive collaboration (i.e., beyond discussing problem set questions) can result in honor code violations. Questions regarding acceptable collaboration should be directed to the class instructor prior to the collaboration. It is a violation of the honor code to copy problem set solutions from others, or to copy or derive them from solutions found online or in textbooks, previous instances of this course, or other courses covering the same topics (e.g., STAT 394/5 or probability courses at other schools). Copying of solutions from students who previously took this or a similar course is also a violation of the honor code. Finally, it is worth keeping in mind that you must be able to explain and/or re-derive anything that you submit.

Violations of the above or any other issue of academic integrity are taken very seriously, and may be referred to the University Disciplinary Boarsd. Please refer to the Allen School's Academic Misconduct webpage for a detailed description of what is allowable and what is not.

Accommodations:

- **Disability Accomodation Policy**: See here for the current policy.
- **Religious Accommodation Policy**: See here for the current policy.

Acknowledgements: Syllabus wording largely influenced by Lisa Yan, Chris Piech, and Mehran Sahami from Stanford University's CS 109, and Anna Karlin's offering of CSE 312.