CSCI 4030/6220 Randomized Algorithms

Course Logistics

Alex Gittens
Department of Computer Science, Rensselaer Polytechnic Institute

Fall 2026

Course Logistics

Instructor:	Alex Gittens (he/him), Lally 316 `gittea@rpi.edu` Office hours: Thu 3–5pm ET, Lally 316
Web:	`https://cs.rpi.edu/~gittea/old-site/teaching/fall2026/randalg/`
Piazza:	`https://piazza.com/rpi/fall2026/40306220/home`
Submitty:	`https://submitty.cs.rpi.edu/courses/f26/csci4030` and `.../csci6220`
Text:	Mitzenmacher & Upfal, *Probability and Computing*, 2nd ed.
Prerequisites:	CSCI 2200 + 2300. Proofs, counting, discrete probability, algorithm analysis, basic linear algebra.

What this course is about

Randomness is an algorithmic resource.

We will use it to do three things:

- design faster algorithms,
- analyze algorithms more cleanly,
- and sometimes replace exact answers with useful approximations.

The course starts with concrete examples, then builds the probability tools needed to analyze them.

Preparing for Each Lecture

- Read the assigned material before the corresponding lecture.
- Use lecture to check your understanding, resolve points of confusion, and encounter the material from a different perspective.
- Come with questions from the reading. This is one of the best ways to use class time.
- Quizzes and exams may cover lecture material, assigned reading, and homework problems.
- In-class quizzes may also ask about the assigned reading for that lecture.

The reading is not a substitute for lecture, and lecture is not a substitute for the reading. Use both.

Assessment

Component	4030 (undergrad)	6220 (grad)
Weekly homeworks (12 total)	50%	40%
Pop quizzes	25%	20%
Final exam	25%	20%
Research presentation	—	20%

Assessment

Component	4030 (undergrad)	6220 (grad)
Weekly homeworks (12 total)	50%	40%
Pop quizzes	25%	20%
Final exam	25%	20%
Research presentation	—	20%

Key dates:

- Homework is assigned weekly.
- Research presentations: Lec 27–28 (6220 only).

Assessment

Component	4030 (undergrad)	6220 (grad)
Weekly homeworks (12 total)	50%	40%
Pop quizzes	25%	20%
Final exam	25%	20%
Research presentation	—	20%

Key dates:

- Homework is assigned weekly.
- Research presentations: Lec 27–28 (6220 only).

Collaboration policy: You may discuss ideas freely, but your write-ups must be your own. Cite any external sources you use, including AI tools.

Assessment

Component	4030 (undergrad)	6220 (grad)
Weekly homeworks (12 total)	50%	40%
Pop quizzes	25%	20%
Final exam	25%	20%
Research presentation	—	20%

Key dates:

- Homework is assigned weekly.
- Research presentations: Lec 27–28 (6220 only).

Collaboration policy: You may discuss ideas freely, but your write-ups must be your own. Cite any external sources you use, including AI tools.

Lecture pace: Some lectures will be denser than others.

Why CSCI 2300 Is a Prerequisite

Algorithmic structure remains central

What you bring from 2300

- Identify the structure of a computational problem.
- Exploit that structure to design an efficient deterministic algorithm.
- Analyze the time and space used by the algorithm.

What changes here

- We use the same problem-solving mindset.
- Randomness becomes an additional algorithmic resource.
- This creates new tradeoffs among correctness, accuracy, time, and space.

The important questions

What structure can we exploit, and what does randomization let us do that a deterministic approach cannot do as efficiently?

Why CSCI 2200 Is a Prerequisite

Probability lets us analyze what randomization buys us

You should already be comfortable with

- basic probability and counting;
- independence and dependence; expectation and variance;
- named probability distributions and their properties;
- conditioning, conditional expectation, and the tower/total-expectation identities;
- tail-sum bounds and related basic tools.

References

- Mitzenmacher–Upfal, Chapters 1–2; or
- *Discrete Mathematics and Computing*, Chapters 15–21.

We will generally use this material without rederiving it in lecture. Questions are always welcome in office hours.

As the course progresses: we will extend this toolkit whenever an algorithm requires stronger probabilistic tools.

Questions?