

## Course Information

**Instructor:** Chanathip Namprempre. Email: `chanathipn at reed edu`.

**Class periods:** MW 13:10 - 14:30 CHEM 301

**Access:** The course website is at <https://nchanath.github.io/387-F22/> Please check the syllabus before every class period.

**Course description from the course catalog.** Introduction to models of computation including finite automata, formal languages, and Turing machines, culminating in universality and undecidability. An introduction to resource-bounded models of computation and algorithmic complexity classes, including NP and PSPACE, and the notions of relative hardness and completeness. Prerequisites: Computer Science 121 or equivalent and Mathematics 112 and 113. Lecture-conference. Cross-listed as Mathematics 387.

**Resources:** The following textbook is required for this course:

- “Introduction to the Theory of Computation” by Michael Sipser. ISBN-13: 978-0534947286. I only have access to the first edition (1997) at the moment.

A useful resource is the following github repository containing Jupyter notebook Python code for exploring string and language operations and simulating various machines: <https://github.com/ganeshutah/Jove>. The code accompanies a textbook entitled “Automata and Computability: A Programmer’s Perspective” by Ganesh Lalitha Gopalakrishnan. You may find the Jupyter notebook code useful for experimenting with concrete examples.

### Evaluation:

- Questions from readings: This portion is worth 10 points. More details can be found at the end of this handout.
- Class participation: We will do problems and hold discussions in class. This portion is worth 10 points.
- Quizzes: There will be five in-class quizzes in this course. This portion is worth 50 points.
- Final: The final exam will cover *all* of the material covered in this course. This portion is worth 30 points.

**A word about writing.** In general, communication skills are very important in computer science. In this course, it is especially important that you learn how to not only communicate clearly but precisely and verifiably as well. In particular, we will be writing mathematical proofs in this course. Having an intuition as to why a statement being proved holds is very important but is insufficient. One goal in the course is to learn how to write proofs that are well-structured and can be verified by readers with mathematics or computer science background.

## Additional instructions

**Questions from readings.** Before the start of each class period, you must submit a list of at least *three* questions that you have as a result of going through the reading material specified for that class period. Consult the course syllabus on the course website for the reading assignment for each class period. Be sure to do so every week as the syllabus is subjected to changes as we go through the course. For each question that you submit, you must clearly classify it into either of the following types:

- Type a: You have this question while doing the reading, but you arrive at an answer afterwards. You can specify the answer, but you are not required to do so.
- Type b: The reading assignment has not answered this question for you even after having completed it.

The questions that you submit should demonstrate that you have done the reading *actively* and have pondered about the material in the reading assignment. The more active the indication, the better the grade. So, while it is possible that you really do wonder how a new concept covered in the reading assignment can be applied in real life, this kind of question is not an effective demonstration of your thoughtfulness. A question of the type “what does the term X mean” is also equally ineffective.

In general, the more specific your question is, the better. It is natural and healthy to be confused when we encounter new concepts. Being able to articulate your confusion is a major enabler of intellectual growth. The questions should be written in your own words since they are supposed to reflect your own thinking process.

This part of the evaluation is evaluated on a 4-level scale: 0 (missing), 1 (need improvement), 2 (fair), and 3 (good). A submission that contains fewer than three questions will receive a grade no greater than 1.