CIS 301 – Theory of Computation
Fall 2020

Textbooks*:** Required: "Introduction to the Theory of Computation 3rd Edition"***
by Michael Sipser ISBN-13: 978-1133187790 ISBN-10: 113318779X
*Recommended: "**Automata, Computability, and Complexity**" by Elaine Rich.*

Instructor: Supraja Gurajala,
Phone: 315-267-2091
Email : gurajas@potsdam.edu

Office Hours: MTWTh: 9:00 am - 11:00am
Discord Text Channel : dr_gurajala_office
Discord Voice Channel : Dr. Gurajala_office

Class Time/Place: MWF 11:30am - 12:20pm, Zoom link provided on moodle course page

Final Exam: Tuesday, Dec. 15, 8:00 - 10:00 am

Course Description:

Theory of Computation1, Regular and context-free languages, Turing machines, and the halting problem. Prerequisites: CIS 203, CIS 300.

Course Overview:

Computer programming is a practical art but the field of **computer science** is firmly grounded in mathematical /theory/. This course examines the theory underlying how computers work. A "cocktail-party" explanation of the content of this course is that we will attempt to describe the very simplest computer that does interesting computations and examine the range of computations it can do. This will require us to formally define what a computation is, what computations are "interesting", and then, finally, what a simple computer is.

We begin by examining the formal definition of a **language** and how computation can be modeled by deciding what strings are in (or out) of a given language. Consider, for example, addition: the string "1+1=2" is in the language of addition; "2 + 2 = 7" and "aardvark" are not. Computation is the execution of a decider; a binary, digital, general-purpose computer is an electronic decider (exact method of operation to be determined).

A hierarchy of different **classes** of languages have been identified. This course focuses on the regular, context-free, and decidable languages. These classes have wide applicability across computer science: regular languages in lexical analysis and text search; context-free languages in parsing programming languages; decidable languages in describing what is and **what is not** computable. Proving that a given language is (or is not) in a particular class is useful in

distributed systems, parallel programming, programming language design, and compiler implementation.

This course will focus on grammars and automata and their use in classifying languages. There will also be an examination of reducibility and computability in terms of languages.

Learning Objectives:

Students who have taken this course should be able to gain:

- knowledge of discrete and continuous mathematics – including elementary probability and statistics – and the ability to apply logic and mathematical proof techniques to computing problems.

  This course covers **languages**, sets of **strings** which are sequences of **characters** draw from some set of characters. These mathematical structures are the basis of the theory of computation. Students will be able to describe languages in multiple ways. Students will be able to write proofs using sets of strings and sequences of characters.

- knowledge of basic theory of computability and complexity of computation.

  Students will be able to construct language deciders from a description of a language, describe the language accepted by a given decider, and prove languages are or are not of a particular class of languages. Students will be able to define and apply computability, decidability, and acceptability in terms of automata and languages.

- knowledge of fundamental data structures and algorithms – including analysis of their correctness and complexity – related to various fields of computer science, and the ability to apply this knowledge to problems through the use of appropriate programming languages.

  Students will be able to define **algorithm** in terms of Turing machines. Students will be able to define what it means to be computable. Students will be familiar with proof by reduction.

Learning Outcomes:

Upon completing this course, students should be able to

1. Build a deterministic finite automaton (DFA), non-deterministic finite automaton (NFA), and/or regular expression (RE) for a given regular language.

2. Prove the equivalence of DFA, NFA, and RE in the set of languages they can accept; prove standard closure properties for regular languages.

3. Prove that a given language is/is not regular (apply regular pumping lemma).

4. Describe the language accepted by a given DFA, NFA, or RE.

5. Build a push-down automaton (PDA) and/or context-free grammar (CFG) for a given context-free language.

6. Prove the equivalence of PDA and CFG in the set of languages they can accept; prove standard closure properties for context-free languages.

7. Prove that a given language is/is not context-free (apply context-free pumping lemma).

8. Describe the language accepted by a given PDA or CFG.

9. Build a Turing machine (TM) for a given language.

10. Describe the language accepted by a given TM.

11. Describe the relationship between regular, context-free, and recursively-enumerable languages.

12. Describe the universal TM, its operation, and how it connects to physical computers.

13. Define and apply the term algorithm.

14. Connect and discuss the relationship between computable, decidable, and recognizable in terms of TM.

Grading for the Course:

1. ***Daily Quizzes: 10 %***
   A daily quiz will be given at the start of each lecture. This will also count as your attendance.

2. ***Weekly Quizzes: 10%***
   A ten-minute weekly quiz will be given starting the second week of classes. It will be based on lectures and exercises (at the end of each chapter).

3. ***Homework / Programming Assignments: 20 %***

   Several programming assignments and homeworks will be given based on the concepts discussed in lectures. These programming assignments and homeworks will be the essential part of the course. Programming assignments and homeworks will be posted on the moodle page along with the due date. Late assignments are penalized at 20% per calendar day that they are late. Your final submitted assignment should represent your individual work; it is, however, acceptable to discuss the solution approach with other students. You will be responsible for keeping track of due dates posted on moodle. Assignment submission policies are detailed in a separate section of this document.

4. ***Exams: 60%***
   a. Exam 1 – 14%
   b. Exam 2 – 14%
   c. Exam 3 – 14%
   d. Final Exam – 18%

   Exam will be closed book and closed notes. Any request for re-grading must be received in writing and within 3 days of receiving your graded exam back. Prior notice must be

given to your instructor in case of emergency. No make-ups will be granted unless satisfactory documentation is produced to show an extenuating circumstance.

Final exam is cumulative and is on Tuesday, Dec. 15, 8:00 - 10:00 am

At the end of the semester I will calculate what fraction of the possible points you have earned, and your grade may be based on this distribution:

90% >=	A
80% - 90	B
70% - 80	C
60% - 70	D
< 60%	F

Note that final grades are determined using a class curve of the course-grade averages.

Due Dates

All due dates for the course will be strictly enforced. Prior approval will be required from the instructor for any late submission.

5. Technical Requirements Summary

Hardware*:* The course is being taught virtually, with all participants working remotely. That means that you will need to have the following computer hardware:

- Laptop or desktop computer – This is a programming-intensive course. You will need a computer to be able to do the programming. If you have only a tablet or a smartphone, please contact me so we can talk about alternatives for you to do the work.
- Camera and microphone – You need these to support video/audio for synchronous class meetings and for using the CS Department Discord server (more information below). Your laptop or desktop system may have built-in camera and microphone, or you could use eternal camera and microphone. You can also use a tablet or smartphone for video/audio communication.

Software*:* Here is a summary of the various software you will need for the course, in addition to the basics of a computer, browser, and typical software.

- **VPN (virtual private network) software** – You may want to connect to the university's VPN so that you can connect remotely to the CS lab in Dunn 302. If you are using Windows or Mac OS, you can find instructions for the software download and setup here: https://www.potsdam.edu/about/administrative-offices/computing-technology-services/services/vpn. If you use Linux, Dr. Ladd has made a video to help you set up to use the VPN. The video is available near the top of the Moodle course page.
- **Command line interface (cli) tool** – If you access the CS lab remotely, you will need a command line tool to work on the lab machines. You will not have access to any graphical user interfaces when working remotely. Windows, Mac, and Linux operating systems have a version of the command line interface available to users.

- **VSCode** – We recommend that you install this free programming environment. It is free (as just noted), available for any OS, easy to use, and allows for users to share code. You can download VSCode from https://code.visualstudio.com/download.
- **Java 11** – This is the version of Java that is installed in the lab, and VSCode will want you to use this version as well.
- **Discord** – The CS Department has Discord server (more information below) that is our "virtual department". My office hours will take place in Discord, our CS tutors will work on Discord, and our ACM chapter has its meetings on their Discord server. You can join our server at https://discord.gg/Np2NEQ and find information about getting started with Discord at https://discord.com/new
- **Zoom** – Our synchronous (real-time) class meetings will take place through Zoom. You can get a free Zoom account here https://zoom.us/. There may be additional updates about Zoom; the university and SUNY are working on finalizing license and technical details.

6. Tentative Schedule

Week 1	Introduction, Mathematical Notions and Terminology, Definitions,
Week 2	Theorems and Proofs , Regular Languages, Finite Automata
Week 3	Non deterministic Finite Automata, Regular Expressions
Week 4	Non Regular Languages, Context Free Grammer
Week 5	Pushdown Automata, Non context Free Grammer
Week 6	Review & Midterm 1
Week 7	Deterministeic Context Free Languages, Turing Machines
Week 8	Variants of Turing Machines, Definitions of Algorithms
Week 9	Decidable Languages
Week 10	Review & Midterm 2
Week 11	Undecidability
Week 12	Reducibility, Undecidable problems
Week 13	Simple undecidable Problem
Week 14	Advance topics in Computability Theory
Week 15	Recursion & Review

Impact of extracurricular activities on class work

You make the choices about how you will spend your time, including investing your time in non-academic activities. As a student, you need to give priority to your academic work, and prevent extracurricular commitments from negatively impacting your work for classes. You are, of course, free to participate in activities that are meaningful to you; however, do not expect me to give special consideration because of time management issues that arise from those activities. You should not be missing class because of extracurricular activities, nor should you allow yourself to fall behind on assignments. **NOTE: I will not give extensions that relate to participation in extracurricular activities, even if the activity is related to Computer Science.**

Computer Use

You can do your programming assignments on the PC workstations in DUNN 358. They are connected via a Local Area Network to a server, so you can sit down at any of the machines and log in to your own account. These machines use a version of the Unix operating system called Linux. We will cover the basics of Unix during the first lab. Many students successfully install Java on their home computers to write, compile and run programs.

You must use your SUNY Card to enter the Unix lab. Your computer account in the Unix classroom is your private property, and should be treated as such. Please protect yourself by keeping your password private and making sure that you completely log out every time you use your account. It is your responsibility to prevent others from plagiarizing your work.

Expectations for the Course

- You are expected to ask questions and participate in class discussions. However, talking out of turn, engaging in non class-related discussions with others (including the instructor), and other disruptive behavior will not be tolerated. If you are disrupting class, you may be asked to leave. Repeated offenses could result in your referral to the Office of Student Conduct and Community Standards and possible dismissal from the class.
- In laboratory, you are expected to collaborate actively with your assigned partner, to follow instructions, and to take turns controlling the keyboard. Failure to meet these expectations during scheduled lab sessions could result in a reduced lab grade (possibly 0) and possibly being dismissed from the laboratory session.
- Academic dishonesty: Students are expected follow the "SUNY Potsdam Academic Honor Code" (SUNY Potsdam 2014-2016 Undergraduate Catalog, p. 42) by doing their own work on quizzes, exams and programming assignments unless specifically directed otherwise by the instructor. Copying is strictly forbidden. Students caught cheating will receive a grade of 0 for that evaluation. Repeated offenses will result in dismissal from the course and possible disciplinary sanctions by the university. Academic Misconduct definitions, procedures, due process, and student rights are described on page 43 of the SUNY Potsdam 2014-2016 Undergraduate Catalog.
- Disability Assistance: Anyone who has special needs that must be accommodated to fulfill the course requirements should notify the instructor and the Director of

Accommodative Services, 111 Sisson Hall, 267-3267. The college has resources available to assist qualified students with their academic studies.

- Food and Drink in Class and Lab**:** Beverages are allowed in the classroom as long you clean up after yourself and do not disturb others. In the Unix lab, food and drink are restricted to the coffee table. UNDER -NO- CIRCUMSTANCES ARE FOOD AND BEVERAGES (EVEN GUM) ALLOWED NEAR THE COMPUTERS.
- No devices are allowed during class. Notes must be hand-written
- Accommodation of Religious Observances: We will make reasonable accommodation for a student's religious beliefs. Please notify us within the first week of classes about any scheduled class date that conflicts with a religious observance.

Assignment Submission Policy and Guidelines

- You will receive a new assignment every week or so. Assignments are to be submitted by the beginning of lecture on the due date. Assignments will be submitted electronically on moodle.
- Late assignments will be penalized by 20% per calendar day that they are late. Extensions that are not subject to penalty may be granted in rare cases when there are extenuating circumstances (such as serious illness or disability, a death in the family, an accident, etc.) and when these circumstances are supported by written documentation.
- Every programming assignment that you submit must have an id-box header like the one shown below.

```
/*
Name: John Doe Course: CIS 301 Theory
Assignment: 1
*/
```

- Work that is not submitted electronically cannot be graded and will receive a grade of zero.
- Programming assignments should reflect your ability to program. We encourage you to talk to others in the class about assignments to discuss approaches to solving programming problems. It is also permissible to look at someone else's code and point out a silly syntactic error. It is also permissible to discuss a programming concept, in general, (for example, if-statements) if it appears that a fellow student does not understand that concept. However, there should be no collaboration on assignments above and beyond this. Specifically,
- You should not be sharing any of your code with anyone else.
- You should not be accepting any code from anyone else.
- Two or more students should not be writing code together so as to essentially produce the same program

- If a tutor is assisting you, that tutor should not be making any suggestions about your code except for general strategies for solving the problem, explaining general programming concepts, or catching simple, syntactic errors as described above.
- If we suspect that students and/or tutors are engaging in unfair collaboration as described above, we reserve the right to call all parties involved in for a code review. If the code review reveals that one or more persons is involved in providing or receiving unfair help, all parties involved will receive a zero on that assignment. Any further problems in this regard with any student or tutor will be brought to the attention of the Dean of Students.
- Since the load on the system and the printer may be very high around the submission deadline, you are urged to get your assignment finished and printed early.

SUNY Potsdam Department of Computer Science Code of Professional Conduct

1. ***Preamble***
   All members of the ACM, including the Computer Science faculty of SUNY Potsdam, are committed to ethical professional conduct as specified in the ACM Code of Ethics and Professional Conduct. Students, taking courses from the faculty, are bound by our commitment.

   All members of the Department are obliged to remind one another to behave professionally. Violations should be reported promptly; however, capricious or malicious reporting of violations is, itself, a violation. When reporting, bring all relevant aspects of the incident to the faculty's attention.
2. ***Moral Imperatives***
   As a Computer Science student I will...

   2.1. Respect all members of the Department.

      2.1.1. Be professional in face-to-face and electronic interactions.

      2.1.2. Be fair so everyone is free to work and learn.

      2.1.3. Be active in preventing discrimination in physical and electronic spaces frequented by Department members.

   2.2. Accept and provide appropriate feedback.

      2.2.1. Avoid starting or spreading rumors.

      2.2.2. Respect confidentiality.

   2.3. Be honest, trustworthy, and respect intellectual property.

      2.3.1. Only take credit for my own work.

      2.3.2. Respect the privacy of others.

2.3.3. Access computing resources only when authorized and report any access risks discovered.

2.4. Contribute to society and human well-being.

2.4.1. Improve public understanding of computing and its consequences.

2.4.2. Consider both the direct and indirect impacts of my actions.

Based on the ACM Code of Ethics and Professional Conduct, retrieved from https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct 11 August 2017

Student Support

Every student in this class is a valued individual. If you are struggling with issues outside of the classroom, please know that there are professionals both on and off campus who can assist you. If you need immediate assistance, please contact our campus Counseling Center (with free counseling) at (315) 267-2330 or visit their website. Links to other resources are provided below:

- • Stacey L. Basford- Title IX Coordinator ▪ Van Housen Extension 392
  ▪ (315) 267-2516
  ▪ basforsl@potsdam.edu
  ▪ http://www.potsdam.edu/offices/hr/titleix

- • Bias Incident Reporting-
  ▪ http://www.potsdam.edu/about/diversity/biasincident

- • Center for Diversity
  ▪ 223 Sisson Hall
  ▪ (315) 267-2184
  ▪ http://www.potsdam.edu/studentlife/diversity

- • University Police
  ▪ Van Housen Extension
  ▪ (315) 267-2222 (number for non-emergencies; for an emergency please dial 911)

- • Student Conduct and Community Standards
  ▪ 208 Barrington Student Union
  ▪ http://www.potsdam.edu/studentlife/studentconduct/codeofconduct

- • Reachout (24-hour crisis hotline) ▪ (315) 265-2422

- • Renewal House (for victims of domestic violence)
  ▪ SUNY Potsdam Campus Office: Van Housen Extension 390 (open Wednesdays, 9-5:00)
  ▪ (315) 379-9845 (24-hour crisis hotline)

- Renewalhouse_campus@Verizon.net

And please: if you see something, say something. If you see that someone that you care about is struggling, please encourage them to seek help. If they are unwilling to do so, Care Enough to Call has guidelines on whom to contact. Everyone has the responsibility of creating a college climate of compassion.