BOB JONES UNIVERSITY
LEARN. LOVE. LEAD.

CpS 440
Theoretical Foundations Of Computer Science
Spring/2025

College of Arts and Sciences
Division of Mathematical Sciences

Professor:	James A. Knisely, Ph.D.
Office:	Alumni 64
Office Hours:	MWF 7:45-8:50 a.m. TTH 7:45-9:15 a.m. Other times by appointment.
Email:	jknisely@bju.edu
Telephone:	Extension 8144
Communication Policy:	For class questions that all students might benefit from, please use the class specific MS Teams team. For other types of questions or notifications, please use the chat feature of MS Teams or email. Most questions involving short answers are responded to within four hours, others within 24 hours. Please email or message if you desire a meeting so that a location can be agreed upon that allows for privacy, help.
Classroom:	AL 314
Meeting:	Lecture meets TTH 2:00 - 3:15 p.m.
Credit/Load:	3/3
Textbook:	*Introduction to the Theory of Computation* by Michael Sipser. Third edition

Catalog Description:

A study in finite state machines, Turing machines, computability, formal languages, etc.

Course Context:

This course serves as a major intersection between Computer Science and Mathematics, therefore, it fullls the following objectives from each department.

Computer Science

- CS1. Design and implement efficient solutions to problems in various domains.
- CS2. Demonstrate understanding of fundamental concepts in computer science.
- CS3. Communicate technical information eectively, including software design and requirements documents.
- CS4. Evaluate and assess software technologies for use in solving specic problems.
- CS5. Apply Biblical principles of ethics to computing.

Math

- DM1. Mature the student in the theory and applications of mathematics.
- DM2. Provide the student the required mathematical background to function and contribute eectively in todayâ€™s technological society
- Provide the student a platform for continued learning and development of his Godgiven abilities.
- Instill in the student a desire to use his abilities in service to Christ.
- Provide an appropriate liberal arts complement to a wide variety of majors

Course Goals:

The goals of this course are to

- improve your knowledge and experience with
  - Template and/or generic functions and classes,
  - Generic and template implementations of data structures in a variety of languages,
  - The Linux operating system, the Windows subsystem for Linux, and/or the Mac terminal
- increase your knowledge of computer science, specifically in these areas:
  - Data structures -- both fundamental and advanced;
  - Recursive algorithms, greedy algorithms, and dynamic programming;
  - Depth-First and Breadth-First Searching

- increase your awareness of social issues involving computers including privacy and civil liberties.

Course Objectives:

Upon successful completion of this course the student will be able to:

1. Discuss the concept of finite state machines.
2. Explain context-free grammars.
3. Design a deterministic finite-state machine to accept a specified language.
4. Explain how some problems have no algorithmic solution.
5. Provide examples that illustrate the concept of uncomputability.
6. Define the classes P and NP.
7. Explain the significance of NP-completeness.
8. Prove that a problem is NP-complete by reducing a classic known NP-complete problem to it.
9. Determine a language's location in the Chomsky hierarchy (regular sets, context-free, context-sensitive, and recursively enumerable languages).
10. Prove that a language is in a specified class and that it is not in the next lower class.
11. Convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, and between PDAs and CFGs.
12. Explain at least one algorithm for both top-down and bottom-up parsing.
13. Explain the Church-Turing thesis and its significance.

Course Content

- Mathematical Notions and Terminology
- Definitions, Theorems, and Proofs
- Finite Automata
- Nondeterminism
- Regular Expressions
- Non regular languages
- Context-free Grammars
- Pushdown Automata
- Non-context-free languages
- Turing Machines
- Variants of Turing Machines
- Definition of an Algorithm
- Decidable Languages
- The Halting Problem
- Undecidable Problems
- Measuring Complexity
- The Class P
- The Class NP
- NP-completeness

Course Requirements:

The grade for this class will be based upon the following categories:

Category	Points	Description
Tests	300	Three tests are scheduled. They will cover one chapter each.
Final Exam	100	The final will cover chapters 4, 5, and 7.
Homework	80	The problems will be due at each test.
Program	20	One program will be assigned.

Grading Scale

A 10-point grading scale, with 1-digit rounding, will be used.

Percentage	Grade
90% - above	A
80% - 89%	B
70% - 79%	C
60% - 69%	D
Below 60%	F

General Policies:

Class Deportment
Compliance with student handbook policies is expected during class. All class deportment should reflect your intention to pay attention, to be polite, and to be professional. Laptops may be used to take notes and to perform calculations and constructions during class. Please do not use the laptop for other purposes during class since studies have demonstrated that one's student's misuse of a laptop during class tends to diminish the learning of the surrounding students.

Accommodations for Students with Disabilities
Any student with disabilities or any additional needs is encouraged to contact the instructor within the first week of the course to discuss accommodations that may be necessary.

Attendance Policies and Academic Penalty for Absences

- Attendance will be tracked and reported according to the university attendance policy: BJU Policies
  - Students are expected to attend and arrive on time for all scheduled class sessions, including the final exam.
  - Students are to use effective time management in order to meet their class attendance responsibilities.
  - Arriving late or leaving early is marked as a partial attendance. Three (3) partial attendance marks count as an absence.
  - Missing a substantial of class is marked as an absence.
  - For more details and information about chronic illness, please see the Class Attendance Policy on the BJU Policies page.
- Students are responsible for all material and announcements given in class.
- If a student is absent for an exam and has a good reason, the student is to notify the instructor before the exam is covered in the next class. *Make-up exams are given at the professor's discretion. A late penalty may be applied.*

Late Work
The following policy is the standard late policy for courses taught in the Department of Computer Science:
Assignments can receive full credit only if submitted by the prescribed deadline. A 25% penalty will be applied if the assignment is not turned in on time. No credit is possible after one week past the original deadline.

The late penalty may be waived for one late assignment. Please discuss this with the professor if needed.

Academic Honesty
You are expected to uphold the school standard of conduct relating to academic honesty:

- School-wide – The link can be found on the BJU Policies page.
- CpS Department clarifications – What is allowed/disallowed in code submissions.

You must assume full responsibility for the content and integrity of the academic work you submit. The guiding principle of academic integrity is that your submitted work; examinations, reports, and projects must be your own work. You are guilty of violating this policy if you:

- Represent the work of others as your own.
- Use or obtain unauthorized assistance in any academic work.
- Give unauthorized assistance to other students.
- Modify, without instructor approval, an examination, paper, record, or report for the purpose of obtaining additional credit.
- Misrepresent the content of submitted work.

Misrepresenting your work is unethical in any setting. In an academic setting, it is a breach of the university policies. The penalty for cheating is severe. Any student cheating is subject to receive a failing grade for the assignment and will be reported to the Dean. If you are unclear about whether a particular situation may constitute cheating, consult with your instructor about the situation. For this class, it is permissible to assist classmates in general discussions of construction techniques. General advice and interaction are encouraged. Each of you must develop your own solutions to the assigned projects, assignments, and tasks. In other words, you may not "work together" on graded assignments with other students unless instructed to work as a group on a particular assignment. Such collaboration constitutes cheating. You may not use or copy (by any means) anotherâ€™s work (or portions of it) and represent it as your own.

Learning how to use sources appropriately is a vital part of your development as a student. To assist you in this endeavor, the university uses Turnitin, an academic plagiarism checker. Registration in this course constitutes permission for the teacher to submit any or all assignments to Turnitin.

Need Help?
You must seek help when needed because you are the only one who knows when you need it. If you need help, reach out to one of the following ways:

- Teacher – It is always best to seek help in person, either in my office or before class, if time allows. You may also text me or email me in order to set up a time in which to come see me if you have a class or are working during my announced office hours. My door is always open during my office hours. I encourage you to come see me for help.
- Classmates – Studying for tests with other students is helpful. You may work together on the homework and presentations. The Academic Honesty CS-specific link above describes the help allowed for programs.

Copyright Policy: