Syllabus

Course Information

Course Number:	CSCE 430
Course Title:	Problem Solving Programming Strategies
Section:	500, 200 (see supplement for section 200 additional information)
Time:	Lecture: MW 3:00-3:50
	Lab: F 3:00-5:30
Location:	Lecture: HRBB 113
	Lab: EABA 118
	See Canvas for links for online access
Credit Hours:	3

Instructor Details

Instructor:	John Keyser
Office:	PETE 408
Phone:	(979)458-0167
E-Mail:	keyser@cse.tamu.edu
Office Hours:	MW 4:00-5:00
	Other times or by zoom available by appointment

Teaching Assistant:	Adil Rasiyani
E-Mail:	adil.rasiyani@tamu.edu
Office hours:	TBD

Teaching Assistant:	Dhruv Patel
E-Mail:	dhruv414@tamu.edu
Office hours:	TBD

Course Description

This course teaches methods for analyzing fundamental programming problems from a variety of domains and implementing solutions quickly and efficiently. The class will use problems based on competitive programming contests to develop skills in problem analysis, coding, and testing. Solving problems will involve identifying and applying a range of algorithmic solutions, including those dealing with combinatorics, dynamic programming, graphs, numerical calculations, string processing, and geometry, along with other more specialized algorithms.

Course Prerequisites

CSCE 411 or Permission of Instructor

Course Learning Outcomes

At the conclusion of this course, students should be able to:

- Analyze a given programming problem to identify the algorithms needed to solve the problem
- Implement a program, including implementing the basic algorithms needed, to solve specified problems
- Develop test cases that will ensure that implemented programs are robust to a full range of valid inputs.

Textbook and/or Resource Materials

Competitive Programming, by Steven Halim, Felix Halim, and (for the 4th edition) Suhendry Effendy. The book can be bought via the website: https://cpbook.net . The 4th edition is in two separate books, available only in paperback. The 3rd edition is available in e-book (and hardcover) format. We will cover almost everything in book 1, and will cover parts of book 2.

The textbook is considered required, in the sense that we will refer to it and you will be able to make use of it through the course. However, students in the past have been able to go through the course without buying it.

Grading Policy

The course grades will consist of a large number of programming problems of varying difficulty and topics given over the course of the semester. Problems will be assigned in 3 different ways:

- Problems to be completed individually on the student's own time. Generally, individual problems will be assigned each week and due on Saturdays, unless the class agrees on a different time. You should expect to spend several hours every week on your own, working on these problems.
- Problems to be completed within a timed period - lab. In each lab period, students will be given a set of problems to be solved during that lab period. These problems are meant to be challenging, with the time pressure being a significant factor.
- Team problems to be completed within a timed period - lab. During some of the lab portions of the class, students may be teamed up with one or two other students, and together they may be asked to complete problems during the lab period. In this case, all students on the team will receive credit for problems completed. *Note: Team problem solving may be limited or removed depending on needs for personal interaction as the semester goes on.*

For each set of problems, students will be given a "base" number of problems they are expected to solve; there may be more problems given than that number, and students may solve more than the base number, if they are able to do so. The total number of base problems over the course of the semester will be approximately 125. In all cases, completing a problem will mean that the code passes a series of validated test data that the students will not be shown. Each problem will be noted as either complete or incomplete by the given time (either the due date or the end of the timed lab period).

Problems completed after the deadline (i.e. after the submission time for weekly individual problems, or outside of lab time during the timed period) can receive half credit. This is

referred to as "upsolving." Students will have a limited period to upsolve these problems late for half credit; typically this will be one week following the original deadline.

The final grade in the course will be based on the percentage of problems solved by each student (i.e. the number completed divided by the base number). That is, the number completed will be the number of problems completed individually on their own time, plus the number completed individually within a timed period, plus the number completed on a team within a timed period. The number possible will be the number of problems assigned as a base for students' own time, plus the number assigned as a base during the individual timed periods, plus the number assigned as a base during team timed periods. All problems will be weighted equally.

To earn an A grade, a student must also get at least 50% of the base points for the problems to be completed individually on their own time, in every week of the class. This can include points for solving during the week, or later upsolving of the problems.

The grading scale will be:

A = 90 % or greater, **and** at least 50% of available problems in individual weeks
B = 80-89 %
C = 70-79 %
D = 60-69 %
F = <60 %

Late Work Policy

Late work is not accepted (see excused absence information below). The deadlines for submission will be set automatically and are strict. Note, however, that the upsolve period allows for solving problems following the due date.

Course Schedule

Schedule

The following is the expected schedule, including some of the types of algorithms that are expected to be covered each week. However, as the semester goes on, this will likely be adjusted.

Week	Topic	Chapters
1	Introduction; Problem formats; Online judging systems; Parsing Input; Formatting Output	1
2-4	Fundamental Data Structures and their implementation on your own or in libraries (various trees, sets, graphs, search structures)	2
5-7	Applying Divide and Conquer, Greedy, and Dynamic Programming Approaches	3
8-11	Applying Graph Algorithms (search, shortest path, minimum spanning tree, network flow, bipartite graph matching)	4, 8.4-8.5
12	String Processing (editing, edit distance, subsequences, suffixes)	6

13	Applying Numerical algorithms and Combinatorics (GCD, LCM, Chinese Remainder Theorem and modular math, Large number computations, generating and counting permutations and combinations)	5
14	Applications of Geometric Algorithms (2D line segment and polygon queries – intersection, area; calculations on a sphere; 3D volume calculations; ray-surface intersection; convex hull; spatial subdivisions)	7
?	Selected additional algorithms and their application (if time allows)	8-9

Optional Course Information Items

Fee for online system

We will be using the Kattis system (see Online Judging below) under a program that allows free use for our educational activities. However, at some point in the semester there may be a fee required to continue to use the Kattis system. This fee will not exceed $35 per student. If this fee becomes required, students must agree to pay the fee individually (likely requiring a credit/debit card payment) promptly.

Online judging

Online judging systems will be used to perform the testing and acceptance of solutions to problems. This will require students to obtain a user account on these external systems. The Kattis system (tamu.kattis.com) will be the main system used, but others might be used on occasion. Students are expected to keep their login information private, as they would for a department computer system. The online system's judgment will be the sole factor determining whether problems are accepted or not, and this will generally mean passing all test cases (not just some) for the problem.

Communication

We will use Canvas for the course materials. The discussion area of Canvas will be used to allow students to post and answer questions, to make course announcements, etc. Grades will be posted on Canvas, though the actual submission and grading will be through the Kattis system. Students are responsible for checking Canvas regularly for communications and occasionally verifying that grading matches their Kattis judgments.

Limitations on Anonymity

The course will use a competitive programming framework for judging and posting results of all problem sets. This means that students will be able to see an indication of which problems other students have attempted, and their success or lack of success in having these accepted. While there can be the option of hiding the specific user name, students cannot expect their performance to be kept completely anonymous from others, and it may be that other individuals can determine their performance on problem sets either by process of elimination, or from other information they determine from the competitive system.

Source Code Presentations

Some problems worked on may have multiple solutions, and throughout the course, individuals' work might be used as examples for illustrating approaches to solving a problem,

writing code, etc. Students may be called on to describe their own solution to a problem or the approach they tried; advance warning will be given in such cases with students given the opportunity to opt-out of presenting. Alternately, students' code may be used as an example shown to other students for how a given solution might be coded; code presented this way will be kept anonymous unless the student has first agreed to let it be presented.

Computers

You will need to use your own computer for programming and submitting assignments, including during lab. Students will need to bring their own laptop to the Friday lab sessions. The laptop should be set up with necessary compilers/software, or access to shared/online sources, sufficient to enable to students to write and test their own computer programs.

University Policies

Attendance Policy

The university views class attendance and participation as an individual student responsibility. Students are expected to attend class and to complete all assignments.

Please refer to Student Rule 7 in its entirety for information about excused absences, including definitions, and related documentation and timelines.

Here are clarifications regarding how excused absences will be handled. Students missing a timed individual or team event for an excused reason will not have the base score for that period used in their grade calculation; any problems upsolved later for half credit will add to both the base and the solved portions of the grade. For the individual problems that students do on their own time (i.e. weekly problem sets), it is expected that students will work on these problems throughout the period of time they are assigned, so that any absences of less than 2 days should not affect the student's ability to complete the problems on time. If the student has an excused absence of more than 2 days during the period for which the problems are assigned, the student will be given a number of days equal to the length of the excused absence, less two days, following the original due date or the student's return (whichever is later). For example, a student who is sick for 4 days during a week would have the deadline for individual problems extended by 2 days. Note that the period for upsolving problems after the deadline for half credit might not be extended, since coded solutions might be posted at that time.

Makeup Work Policy

Students will be excused from attending class on the day of a graded activity or when attendance contributes to a student's grade, for the reasons stated in Student Rule 7, or other reason deemed appropriate by the instructor.

Please refer to Student Rule 7 in its entirety for information about makeup work, including definitions, and related documentation and timelines.

Absences related to Title IX of the Education Amendments of 1972 may necessitate a period of more than 30 days for make-up work, and the timeframe for make-up work should be agreed upon by the student and instructor" (Student Rule 7, Section 7.4.1).

"The instructor is under no obligation to provide an opportunity for the student to make up work missed because of an unexcused absence" (Student Rule 7, Section 7.4.2).

Students who request an excused absence are expected to uphold the Aggie Honor Code and Student Conduct Code. (See Student Rule 24.)

Make-up work will be handled in this course as described above under attendance. Generally, there will not be make-up opportunities for timed activities (i.e. lab activities) but they will also not count in the base score. For the longer problem sets, the same problems will be solved with the deadline extended as appropriate.

Academic Integrity Statement and Policy

"An Aggie does not lie, cheat or steal, or tolerate those who do."

"Texas A&M University students are responsible for authenticating all work submitted to an instructor. If asked, students must be able to produce proof that the item submitted is indeed the work of that student. Students must keep appropriate records at all times. The inability to authenticate one's work, should the instructor request it, may be sufficient grounds to initiate an academic misconduct case" (Section 20.1.2.3, Student Rule 20).

You can learn more about the Aggie Honor System Office Rules and Procedures, academic integrity, and your rights and responsibilities at aggiehonor.tamu.edu.

For this course, a significant amount of work will require solving problems for which a solution or test data might be available or posted online. Unless otherwise specified, students are *not allowed* to seek out or examine code/data for these problems on their own, prior to turning in their own solutions. Likewise, students should not read explanations of how to solve a specific problem prior to the original due date (i.e. the original deadline); note that reading a **description** of the solution **is** allowed *for problems solved after the deadline for half credit (i.e. during upsolving)*. Accessing unallowed information will be considered a violation of the honor code, and students caught doing so will be referred to the honor council, regardless of whether the actual code is copied or not. Given the ease with which it may be possible to cheat in this way, any violations should expect to receive the maximum penalty from the honor council.

Americans with Disabilities Act (ADA) Policy

Texas A&M University is committed to providing equitable access to learning opportunities for all students. If you experience barriers to your education due to a disability or think you may have a disability, please contact Disability Resources in the Student Services Building or at (979) 845-1637 or visit disability.tamu.edu. Disabilities may include, but are not limited to attentional, learning, mental

health, sensory, physical, or chronic health conditions. All students are encouraged to discuss their disability related needs with Disability Resources and their instructors as soon as possible.

Title IX and Statement on Limits to Confidentiality

Texas A&M University is committed to fostering a learning environment that is safe and productive for all. University policies and federal and state laws prohibit gender-based discrimination and sexual harassment, including sexual assault, sexual exploitation, domestic violence, dating violence, and stalking.

With the exception of some medical and mental health providers, all university employees (including full and part-time faculty, staff, paid graduate assistants, student workers, etc.) are Mandatory Reporters and must report to the Title IX Office if the employee experiences, observes, or becomes aware of an incident that meets the following conditions (see University Rule 08.01.01.M1):

- The incident is reasonably believed to be discrimination or harassment.
- The incident is alleged to have been committed by or against a person who, at the time of the incident, was (1) a student enrolled at the University or (2) an employee of the University.

Mandatory Reporters must file a report regardless of how the information comes to their attention – including but not limited to face-to-face conversations, a written class assignment or paper, class discussion, email, text, or social media post. Although Mandatory Reporters must file a report, in most instances, you will be able to control how the report is handled, including whether or not to pursue a formal investigation. The University's goal is to make sure you are aware of the range of options available to you and to ensure access to the resources you need.

Students wishing to discuss concerns in a confidential setting are encouraged to make an appointment with Counseling and Psychological Services (CAPS).

Students can learn more about filing a report, accessing supportive resources, and navigating the Title IX investigation and resolution process on the University's Title IX webpage.

Statement on Mental Health and Wellness

Texas A&M University recognizes that mental health and wellness are critical factors that influence a student's academic success and overall wellbeing. Students are encouraged to engage in proper self-care by utilizing the resources and services available from Counseling & Psychological Services (CAPS). Students who need someone to talk to can call the TAMU Helpline (979-845-2700) from 4:00 p.m. to 8:00 a.m. weekdays and 24 hours on weekends. 24-hour emergency help is also available through the National Suicide Prevention Hotline (800-273-8255) or at suicidepreventionlifeline.org.

Honors Section Additional Information

Welcome to the Honors section of CSCE 430. This is to explain the difference between how the honors section of the class and the regular sections of the class will operate.

The lecture and the lab portions of the class will run as usual, in conjunction with the rest of the class.

The weekly problem set assignments will be slightly different for you. Through the semester, there will be 15 +/-1 weekly problem sets assigned. In 10 of those weeks, one of the problems will be designated as a non-honors problem, meaning it is not assigned to you. These problems will be identified on the course website each week. Typically, this will be one of the mid-range difficulty problems in the set. Please note that while you may solve these for your own benefit, you will not be able to get credit for them as bonus/extra credit or for upsolving.

You will, instead, be required to create two problems yourself.

- Each of these problems will count as 5 solved problems for determining your grade.
- You must submit these problems to pass the class with more than a C. If you do not submit both problems, your maximum grade in the class (regardless of all other items) will be a C.
- One problem is due Sunday, March 20 (i.e. end of Spring Break)
- A second problem is due Tuesday, May 3 (last day of class)

Here is what is required from you for your problem creation:

- You should create a problem that meets the requirements of the Kattis Problem Format. Detailed links to the format are below, but generally this means including:
    - A problem description (.tex file giving the problem description, input format, and output format)
    - One or more sample inputs, with corresponding outputs
    - Multiple "secret" inputs used to **thoroughly** test your problem, with corresponding outputs. You must include tests that explore all edge cases, the extremes of input, typical inputs, and different patterns of input (when appropriate). Typically this will mean writing a program that can generate test cases.
    - At least one correct solution to the problem
    - An input verifier that can be used to verify that an input matches the requirements
    - A metadata file
    - If desired (these are NOT required), you may include:
        - More than one correct solution program
        - Examples of solutions that should return wrong answer, run-time error, or time-limit exceeded results (useful especially if you want to ensure only efficient implementations are accepted)
        - An output validator (necessary if the output could have multiple correct answers or formats)
        - Hint, Description, or Illustration files for the test data.
    - The problem should pass the Kattis verification tools.
- A short writeup (for me) that describes the problem design
    - I expect this will be ½ page to 1 page in length

    - You should identify what the goal of the problem is, in terms of what concept(s) you are wanting to illustrate/test with the problem. For instance, if the problem is meant to involve a shortest-path algorithm, say so. If it's meant to provide lots of corner cases, say that. If it's meant to just involve a lot of programming that's straightforward, say that.
    - You should give a very brief explanation of how to solve the problem. That is, give a description for what a person is expected to do so that someone reading the description who is a decent programmer could go ahead and solve the problem.
    - List anyone who helped you with the problem, and in what capacity (see below).
- The expectation is that these problems may be submitted to Kattis for possible use, either in future versions of this class, or for more general use. *If for some reason you do not wish to allow your problem(s) to be used in this way, please indicate so, clearly, in your writeup.*
- At least one of your two problems should be something more than just a direct, simple application of a given algorithm. That is, it should cause the person solving it to have to do more than just implement a straightforward algorithm on a particular input format. Some examples of how this might work:
    - Require someone to implement multiple algorithms together – e.g. combine a graph problem with a combinatorics problem. Or, transform one problem to another type then solve that.
    - Create a problem that, in addition to having a direct solution, has a number of special cases that must be dealt with (possibly individually).
    - Create a problem that requires an interesting and not-very-straightforward mapping from the problem description to an algorithm. That is, create a problem that on first reading might seem like one of type X, but actually requires a solution of type Y.
- At least one of your two problems should exhibit a time-constraint, where a poor choice of algorithm would lead to a Time Limit Exceeded result. For such problems, you should be sure that your problem submission includes a TLE program in the solution section, and that your test cases clearly separate good solutions from those that would give a TLE result
- Your submission should be zipped into a single file:
    - The file should contain the problem writeup and a directory following the Kattis format for the problem information.
    - This one .zip file should be uploaded to Canvas. Note that this is the only time in the class that Canvas will be used for submitting problems.
- To help ensure that you are on the right track, each person will need to meet individually with the instructor prior to the final problem deadline. *At this meeting, you should have a draft of the short writeup describing problem design (see above) available for me to review.* These meetings will be scheduled and held (possibly by zoom, possibly in person) over a 2-week period ending approximately 1.5 weeks before the problem deadline, specifically:
    - For problem 1: Feb. 28-March 11
    - For problem 2: April 11-22

Please note: You should write all the required parts of the problem yourself, including a correct solution and the majority of the test cases. However, if you wish to include additional solutions (e.g. of wrong answers, or of multiple acceptable answers), you can include solutions written by other people. You may also include test cases provided by other people. You are welcome to get feedback from others, have others "try" your problem on their own, etc. before submitting it. In fact, you are encouraged to do so – others will typically be able to identify and point out ambiguities or other issues. They might also be able to help identify cases you did not consider.

I want this to be a fun and interesting experience. I believe you will find that the process of coming up with new problems is both challenging and intellectually stimulating.

A description of the Kattis format is at:
http://www.problemarchive.org/wiki/index.php/Introduction_to_the_Kattis_Problem_Format
With more details provided at:
http://www.problemarchive.org/wiki/index.php/Problem_Format
The Kattis problemtools repository (for verifying the problems) is available from github:
https://github.com/Kattis/problemtools