

Competitive Learning I/II CS 2984/CS 4984

Staff

Instructor: Dr. Godmar Back
VTKW 2212, 1-3046
Email: gback@vt.edu

Meeting times:
Scheduled is TR 5pm-6:15pm HAHN-C 130. Actual meetings TBD.

Office hours: TBA

Catalog Description

Fundamentals of algorithms, data structures, and implementation techniques, taught in a setting that combines collaborative practice with competitive exercise. Students practice to solve problems using a computer, which are judged by automated evaluation software for correctness and efficiency. Problems are drawn from multiple areas in Computer Science. Macro- and micro optimization techniques to improve efficiency are emphasized. Students also practice to qualify for, and represent Virginia Tech at an intercollegiate programming contest.

CS2984: A grade of C or better is required in prerequisite CS 1114.

CS4984: Successful participation in Competitive Learning I as well as a grade of C or better is required in prerequisite CS 2114, or instructor approval. Note that approval is generally reserved for those who did participate in CL I or can prove significant experience in competitive programming, such as qualifying for a regional contest.

Learning Objectives

Having successfully completed this course, students will be able to:

- Read and understand problem descriptions given in a technical and/or mathematical language and derive computational approaches.
- Effectively manage their time in designing, implementing, testing, and debugging their solutions.
- Work in teams with peers to collaboratively solve problems.
- Use acquired skills in communication, resource allocation, and negotiation within teams of peers.
- Select and apply standard data structures, including arrays, lists, hash maps and binary trees as appropriate.

- Select and adapt standard algorithms to a variety of computational problems, including recursion, divide-and-conquer, greedy algorithms, backtracking, and various search algorithms.
- Analyze the algorithm's complexity and choose appropriate algorithms for a given problem budget.
- Recognize and solve graph-related problems such as minimum spanning tree, connected components.
- Use advanced data structures such as segment trees to solve range-minimum queries.
- Use advanced string-related data structures such as suffix arrays.
- Recognize and solve network flow problems.
- Understand and apply dynamic programming as a problem solving technique.
- Solve basic and advanced problems in computational geometry, including 2D and 3D vector geometry.
- Be proficient in applying numerical algorithms for optimization and root finding.
- Use efficient code libraries and techniques for fast input/output (I/O) and apply language-specific micro-optimizations, such as custom memory allocation, to their program code.

Justification

Whereas the mastery of algorithms to solve problems has been a cornerstone of computer science education for many years, there has been a recent surge in interest in programming competitions. Programming competitions are really problem solving contests in which students adopt and adapt algorithmic problem solving techniques and apply them to specified problems drawn from a variety of application areas. These competitions not only reinforce these algorithmic problem solving techniques, but do so in an engaging context. The Virginia Tech Association for Computing Machinery (ACM) Programming Team has grown to be a significant activity for our undergraduate students, and this course will provide a framework for its activities.

Aside from improving problem solving abilities, the problem solving skills acquired in this activity will also help students prevail in challenging technical interviews, thus strengthening their employment prospects for internships and positions.

Texts and Special Teaching Aids

Recommended (but optional) textbooks include:

- Steven Skiena and Miguel Revilla, Programming Challenges, Springer-Verlag New York, Inc., USA, 2003, ISBN: 978-0-387-00163-0 (Print) 978-0-387-22081-9 (Online)
- Steven Halim and Felix Halim, Competitive Programming, 3rd Edition, available on [Amazon](#)

Expectations

- Attend any scheduled class meetings regularly.
- Participate in at least 75% of scheduled practices.
- Participate in Holiday Contest late November/early December.
- Develop one original problem using problemtools well before the VTHS contest, including meeting milestone deadlines for ideation, prototyping, and refinement/finalization.
- Participate in at least 3 outside competitions with original problems (e.g. CodeForces, CodeAcademy, CodeChef, or others.)
- Create solution sketches for 3 problems you have solved this semester using LaTeX/overleaf.
- Be active and engaged.
- Be a good team member.
- If you do not plan on meeting those expectations, please do not enroll in the class. You may still participate in all programming team related activities.

Grading

Despite the title “Competitive Learning,” this is probably the least competitive class you’ll take. Grading is not done based on the performance in the practices or contests, but rather based on meeting the expectations outlined above, which are mostly participation-based (except for the original problem design).

Honor Code

The Undergraduate Honor Code pledge that each member of the university community agrees to abide by states:

“As a Hokie, I will conduct myself with honor and integrity at all times. I will not lie, cheat, or steal, nor will I accept the actions of those who do.”

Students enrolled in this course are responsible for abiding by the Honor Code. A student who has doubts about how the Honor Code applies to any assignment is responsible for obtaining specific guidance from the course instructor before submitting the assignment for evaluation. Ignorance of the rules does not exclude any member of the University community from the requirements and expectations of the Honor Code.

For additional information about the Honor Code, please visit:

<https://www.honorsystem.vt.edu/>

Specific rules for the honor code in this class include:

- Abide in all contest by the stated rules as far as the use of online materials, previously written code, collaboration with others, etc. is concerned. For each contest, these rules will be stated clearly.

- When using code that you have not written yourselves where this is allowed, document this clearly in your submissions. An exception is any code that appears in our team handbook.

Special Accommodations

Students are encouraged to address any special needs or special accommodations with me during the first two weeks of the semester, or as soon as you become aware of your needs. Those seeking accommodations based on disabilities should obtain a Faculty Letter from the Services for Students with Disabilities office (540-231-0858) located in Lavery Hall, Suite 310) <http://www.ssd.vt.edu/>.