

COURSE DESCRIPTION

Department and Course Number	CS 497	Course Coordinator	Gray
Course Title	Competitive Programming Techniques	Total Credits	1

Current Catalog Description

This course will help students to be more competitive in the ACM programming contest by exploring numerous problem solving techniques and algorithms not covered in the traditional curriculum.

Textbook

Programming Challenges: The Programming Contest Training Manual, by Steven S. Skiena and Miguel A. Revilla, Springer-Verlag, 2003.

References *None*

Course Goals

Provide a unique opportunity to explore new problem solving techniques in a way that is not covered traditionally in the curriculum. This should help students add to their knowledge to improve software development skills and help our students to be more competitive at the ACM programming contest.

There are many additional benefits, such as exposure to new algorithms, group programming techniques, and general concerns regarding good design of software in a team environment. The course will have numerous lectures on specific algorithms and techniques. The lectures are to be followed by group efforts that solve previous contest problems from an application of the algorithm/techniques studied. Students will learn to identify typical solution patterns for commonly recurring contest problems (e.g., dynamic programming, parsing). The techniques for efficient utilization of resources within a team environment will also be studied and practiced. Students will be expected to meet outside of class for 2-3 hours each week to work on practice problems that are assigned.

Prerequisites by Topic

Discrete Structures

Major Topics Covered in the Course

Advanced Data Structures; Strings; Sorting and Searching; Backtracking; Dynamic Programming; Standard Template Library; Graph Algorithms; Number Theory

Laboratory projects (specify number of weeks on each)

There are no assignments that could be considered laboratory projects. Each week students are given sample problems to solve from real contest scenarios.

Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	_____	<u>5</u> _____	Computer Organization and Architecture	_____	_____
Algorithms			Concepts of Programming Languages	_____	<u>3</u> _____
Software Design	_____	<u>5</u> _____			

Oral and Written Communications

None

Social and Ethical Issues

None

Theoretical Content

Combinatorics (1 hour)
Number Theory (1 hour)
Graph Algorithms (1 hour)

Problem Analysis

All course sections contain a significant number of sample competition problems that require the students to perform detailed analysis prior to any coding. Pencil and paper analysis is encouraged before moving to a keyboard. The students are often put under stress with time deadlines for each problem to simulate contest situations.

Solution Design

All course sections contain a weekly in-class and homework assignments that require software design to solve the assigned problem. These problems are often from past programming contests. Students are also asked to reveal and explain their analysis to the rest of the class.