

COP 4516: Problem Solving Techniques and Team Dynamics Syllabus

Course Website: <http://www.cs.ucf.edu/~dmarino/progcontests/cop4516/spr2016>

Lecturer: Arup Guha

Email: dmarino@cs.ucf.edu

Class Times/Locations: T 10:30 – 11:20am (HEC-103),

F 9:00 – 11:50am (HEC-202, HEC-308)

Office: HEC – 240

Office Phone Number: 407-823-1062

Office Hours: MWR 10 - 11 am, TR 1:30 - 2:30 pm

I do NOT check my WebCourses email. Please email me at dmarino@cs.ucf.edu to contact me.

This course requires Friday attendance. Please don't plan trips where you leave on Thursday night.

Course Description: This course covers training similar to that given to UCF's programming team. Lectures will cover classical algorithms, some of which are taught in Computer Science 2, that tend to be useful in solving programming contest problems. Emphasis will be placed on implementation issues. The general topics covered are: number theory, brute force search, greedy algorithms, graph algorithms, dynamic programming algorithms and geometry algorithms.

Note: There is NO course textbook. Rather, course notes and websites will be used as primary sources. If one strongly desires a book, here are a few that would suffice:

Introduction to Algorithms – Cormen, Leiserson, Rivest, Stein (ISBN: 978-0-262-03384-8)

Programming Challenges – Skiena, Revilla (ISBN: 0-387-00163-8)

Algorithms – Dasgupta, Papadimitriou, Vazirani (ISBN: 0-07-352349-2)

The Design and Analysis of Algorithms – Levitin (ISBN: 0-321-35828-7)

Competitive Programming 3 - Halim and Halim (cpbook.net)

Grading

This course will have five components

Item	Quantity	Total Percentage
Hackpack	1 (in teams)	10
Online Contest Participation	2	10
Individual Contests	6	30
Team Contests	6	30
Individual Contest Exam	1	10
Team Contest Exam	1	10

Hackpack

Each team will be responsible for producing one hackpack. Minimal requirements will be given for the hackpack. If these are met, 80% will be awarded. The last 20% will be unspecified and based on exceeding the minimum requirements. Each team will make a single submission of their hackpack towards the end of the course, but it's strongly suggested that each team create weekly internal deadlines for the hackpack.

Online Contest Participation

There are several online websites that run programming contests. Some of these are: CodeForces, USACO, CodeChef and TopCoder. You must tell me your login for these websites and compete in two real contests **before** April 26, 2016. If you want to compete on a site other than these, please tell me the site and I'll let you know whether or not I approve it. In order to get full credit for this component of the course, I will require that you get at least one problem correct in contest.

Individual Contests

For the first six weeks of the course, individual contests will be held on Fridays, each with three problems. 100% will be given to any question correctly submitted during the contest. 80% will be given to any question correctly submitted after the contest, before the test data is posted. 60% will be given to any question correctly submitted after the test data is posted. For any submissions that don't work, a maximum of 50% will be awarded, depending on the number of test cases the submission solves correctly.

Team Contests

For the second half of the course, each Friday competition will be in teams. Grading will be determined in the same manner as the individual contests and the same grade will be assigned to each team member, regardless of who solves which questions.

Individual Contest Exam

After the first six weeks of the course, a more comprehensive contest will be given for individuals. The only differences are that no submissions will be allowed after the contest is over, that this contest counts towards 10% of the final grade instead of 5% and all of the questions in this contest will be newly created for the contest itself. Partial credit will be more generous since no submissions are allowed after the contest is over.

Team Contest Exam

During the final exam period of the course, the teams will compete in a final contest, which will be designed to be easily solvable with a good hackpack. No submissions will be allowed after the contest has ended but more generous partial credit will be given to incorrect submissions (up to 80% instead of 50%.) As usual, any problems completed during the contest will automatically receive 100%.

Academic Misconduct Policy

Since this is an elective (you don't have to be here, so I assume you are here because you WANT to be), I will be more harsh with academic misconduct than usual. **In particular, if there are any clear violations of the academic misconduct policy, I will make official documentation with the necessary witnesses, record the transgression with UCF and fail you from the course.**

The rules for the course are as follows:

- 1) During any individual contests, individuals may ONLY look at language APIs online and no other electronic materials. Students may look at any printed materials. Students **may not** talk to any other students during the individual contests about any items that I might think may be helpful in solving the problems. I reserve my right to use my discretion on whether or not a topic of conversation may be helpful in solving a problem. You are safe in telling someone where the bathroom is or describing where a restaurant is located, for example. You are NOT safe in explaining the steps of any algorithm or pointing out a restriction in a problem, for example.
- 2) During team contests, you may only talk to your team members about problem related issues and you may ONLY look at language APIs online and no other electronic materials. You may look at any printed materials. Communication with non-team members in the course is limited as previously described.
- 3) Hackpack – you may look at algorithmic descriptions in print or electronically in developing your team's hackpack, but **ALL** of the code must be written originally by a combination of your team members. It is perfectly fine if some or all of the code is written individually, or if all the code is produced with each team member present. It's likely that some code will be written individually and that teams will meet to “finalize” algorithms in their hackpack periodically so that each team member is comfortable using any part of the hackpack.

Tentative Schedule

Week	Tuesday Class	Friday Class
Jan 11	GCD, Perm, Combos	Ind Contest #1
Jan 19	Greedy	Ind Contest #2
Jan 25	Trees	Ind Contest #3
Feb 1	Graph – DFS, BFS	Ind Contest #4
Feb 8	Shortest Distance, Top Sort	Ind Contest #5
Feb 15	More Brute Force	Ind Contest #6
Feb 22	Exam Review	Ind Contest Exam
Feb 29	Team Dynamics	Team Selection, Hackpack Planning
Mar 7	Spring	Break!!!
Mar 14	Graph – Network Flow	Team Contest #1 Drop Date!!!
Mar 21	DP – Take it or leave it (Knapsack, LCS, Neighbor) (Guest Lecture)	Team Contest #2
Mar 28	DP – Edit Distance, MCM, World Series, Traveling Salesman DP	Team Contest #3
Apr 4	Geometry – Areas, etc.	Team Contest #4
Apr 11	Coordinate Geometry –Intersections	Team Contest #5
Apr 18	Binary Search Applications	Team Contest #6
Apr 25	Exam Review	
May 2	Team Contest Exam (10 am – 1 pm)	

I may change this schedule, thus class attendance is important. This is a general time frame only and is subject to the needs of the class. At the end of each class I will tell you what we will be discussing during the next class period. I may not post formal notes from the lectures, so please take all necessary notes during lectures. Good notes for most of the topics I will cover can be found online. I will make whatever notes to which I have access available online for students.