

**CSCI 310 – Advanced Algorithms**  
**Fall 2008**  
**Syllabus**

**Instructor:** Semmy Purewal, Ph.D. ([www.cs.cofc.edu/~semmy](http://www.cs.cofc.edu/~semmy))  
**Contact:** [semmy@cs.cofc.edu](mailto:semmy@cs.cofc.edu), (843)-953-7542  
**Office:** JC Long 219  
**Office Hours:** MWF 12:00-3:00, also available by appointment

**Class Time:** MWF 11:00-11:50, JC Long 219

**Course Homepage:** [www.cs.cofc.edu/~semmy/csci310](http://www.cs.cofc.edu/~semmy/csci310)

**Text:** *Algorithms* by Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani.

**Course Description (College Catalog):** A course that covers algorithms, focusing on computational complexity, approximation, classification, and optimization. Algorithms covered include evolutionary and genetic algorithms, gradient descent techniques, discrete optimization, branch-and-bound, dynamic and stochastic programming, combinatorial optimization, and approximation algorithms.

**Course Description (My Description):** We'll study some important algorithms and algorithmic paradigms, refine our ability to analyze the efficiency of algorithms, and make attempts to translate some of these abstract algorithms into Perl. We'll also learn about a class of problems that most likely do not admit efficient algorithms, and how to recognize problems in that class.

**Course Prerequisites:** CSCI 230 (Data Structures and Algorithms) and MATH 207 (Discrete Structures). Both of these prerequisites will be relied on heavily! In particular, I'll assume that you're familiar with common data structures (e.g. stacks, queues, lists, trees), that you've coded several of these in a high-level programming language and that you have at least a rudimentary understanding of proof techniques, combinatorics (i.e. counting) and probability.

**Evaluation:** You will be evaluated based on your performance on three exams, homework, programming and in-class activities.

Exams: We'll have three in-class, closed book/closed notes exams that will each account for 15% of your grade. I'll describe the format in class before the exam date. In order to pass the class you must take the final exam.

Homework: A homework problem or two will be assigned almost daily. The write-ups will always be due THE FOLLOWING MONDAY at the beginning of class! So for instance, if two problems are assigned on Monday, and one is assigned on Friday, you will be expected to turn in 3 problems on the following Monday. I will grade these and return them to you by Wednesday. Your average on these assignments will account for 20% of your grade.

Programming: We'll try to learn enough Perl to code up examples of some of the algorithms that we are studying. I'd like everyone to choose at least 3 to code – each of these will count as 5% toward your 'programming' grade. You will be expected to drop by office hours to demonstrate your code to me when you are happy with it. At that time, I'll either give you full credit for it, or make suggestions for improvement. You'll also be expected to present at least one of your programs to the class, which will count as your 'code presentation' grade (10%). If you do not demonstrate an acceptable program to me by the due date, then you will forfeit that 5% of your grade.

In-class Activities: We'll have various unannounced in-class activities throughout the semester. Some of these will have a component that is turned in, and some will not. When you turn something in with your name on it, it will count toward your 'in-class activities' grade. The value will depend on the total number of activities given throughout the semester. The average will account for 10% of your grade.

*This course syllabus is a general plan for the course; deviations announced to the class by the instructor may be necessary.*

Grade Computation: I follow a relatively standard grading scale, meaning 90 guarantees at least an A, 80 guarantees at least a B, 70 guarantees at least a C, 65 guarantees a D, and below 65 is an F. Plus and minus grades may be given out at my discretion. Your numerical grade will be computed as follows:

Exam 1: 15%  
Exam 2: 15%  
Homework: 20%  
Programming: 15%  
Code Presentation: 10%  
In-class Activities: 10%  
Final Exam: 15%

### **Expectations and Policies:**

Attendance Expectations: I expect you to attend every class. If you are not going to be able to make it, I hope that you will drop me an e-mail letting me know. I will not take attendance, but unannounced in-class activities will be regularly given, and they will count toward your grade.

Out-of-class Expectations: I expect you to spend between 6 and 9 hours per week outside of class on this material.

Class Expectations: I expect you to be attentive in class (please no newspapers, crosswords, books, etc.) and that you participate in class discussions. I do not require that you raise your hand, but I do expect that you make an effort to interject at appropriate times, and that you are respectful and willing to yield the floor to other students.

Reading Expectations: I expect that the assigned readings be completed before the class in which the material is covered. Expect to spend at least 2-3 hours per week reading the textbook.

Exam Expectations: I expect you to be present for exams. As a general rule, I do not give make-up exams. If a serious issue arises, however, let me know ASAP and we can work something out. I can often give exams early if you have a college-approved reason that you have to miss them.

Technology Policy: If you would like to use a laptop or any other electronic device during class time, you must discuss it with me first. I prefer that you begin by writing a short note explaining how you will use it and how it will enhance your learning. Then we will come up with a plan whereby you will be held accountable for using it in the agreed-upon manner.

Homework Expectations: You are encouraged to work with other students and me on homework assignments, but when you “write up” your solutions to be turned in, you must do so by yourself. Expect to spend at least 3-4 hours per week on homework.

Programming Expectations: In this class you are expected to write all code on your own! See 'Academic Honesty Expectations' below for more information. Expect to spend 2-3 hours per week programming.

Academic Honesty Expectations: You are expected to abide by the honor code described on page 10 of the student handbook. Note that even though copying computer code from any source (including the internet) is not specifically mentioned, I consider it plagiarism and will deal with it as such. This means that copying computer code is Class 2 violation, and I will request that the sanction be the assignment of an 'XF' for the course. For homework assignments, if you understand the solution that you've submitted for credit and can recreate it without outside help, then you are not violating the academic honesty policy – otherwise, you are.

Special Needs: If you have any special needs, please contact me as soon as possible via e-mail or office hours. I am happy and willing to accommodate all documented disabilities.

*This course syllabus is a general plan for the course; deviations announced to the class by the instructor may be necessary.*

Week Of	Topics	Reading	Due
08/25/08	Aug 27: Expectations, Policies, Computing Environment (Stono, Emacs/VI, Perl) Aug 29: Sorting Algorithms! (Review of Algorithm Analysis)	None	
09/01/08	Sep 1: Algorithms with Numbers (Multiplication, Division) Sep 3: Modular Arithmetic Part 1 (Addition, Multiplication, Exponentiation) Sep 5: Euclid's Algorithm & its Extension	1.1-1.2	
09/08/08	Sep 8: Randomized Primality Testing Sep 10: More about Randomized Primality Testing & Cryptosystems Sep 12: RSA Cryptosystem	1.3-1.4	
09/15/08	Sep 15: Divide and Conquer Algorithms (Multiplication) Sep 17: Recurrence Relations Sep 19: Merge Sort (Quick Sort?)	2.1-2.3	
09/22/08	Sep 22: Medians Sep 24: Catch-up/Review/Problem Solving Sep 26: Exam 1	2.4	Sep 26, Program 1
09/29/08	Sep 29: Graph Representations Oct 1: Depth First Search Oct 3: Depth First Search (continued)	3.1-3.3	
10/06/08	Oct 6: Strongly Connected Components Oct 8: Breadth First Search Oct 10: Breadth First Search (continued)	3.4, 4.1-4.3	
10/13/08	Oct 13: No Classes (Fall Break) Oct 15: Dijkstra's Algorithm Oct 17: Dijkstra's Algorithm (continued)	4.4-4.5	
10/20/08	Oct 20: More general shortest-path algorithms Oct 22: Catch-up/Review/Problem Solving Oct 24: Exam 2	4.6-4.7	Oct 24, Program 2
10/27/08	Oct 27: Minimum Spanning Trees Oct 29: Minimum Spanning Trees (continued) Oct 31: Huffman Encoding	5.1-5.2	
11/03/08	Nov 3: Huffman Encoding (continued) Nov 5: Dynamic Programming Nov 7: Dynamic Programming (continued)	5.2, 6.1-6.3	
11/10/08	Nov 10: Knapsack Nov 12: Chain Matrix Multiplication Nov 14: Shortest Paths Revisited	6.4-6.6	
11/17/08	Nov 17: NP-Completeness Introduction Nov 19: NP-Complete Problems Nov 21: Reductions and Examples	8.1-8.3	
11/24/08	Nov 24: More Reductions and Examples Nov 26: No Classes (Thanksgiving) Nov 28: No Classes (Thanksgiving)	8.3	
12/01/08	Dec 1: Coping with NP-completeness Dec 3: More coping with NP-completeness Dec 5: Genetic Algorithms(?)	9.1-9.3	Dec 5, Program 3
12/08/08	Dec 8: Review/Catch-up/Problem Solving (Last Day of Classes)	None	
<b>Final Exam is Wednesday December 17, 12:00-3:00</b>			

*This course syllabus is a general plan for the course; deviations announced to the class by the instructor may be necessary.*