American University
College of Arts and Sciences
Department of Computer Science
Washington, D.C. 20016

Semester: Fall 2010
Course Number: CSC 520
Title: Algorithms and Data Structures
Prerequisites: CSC281 or equivalent, and able to program in a language such as Python, C, C++ or Java. Students will be expected to design and debug fairly large program.

Instructor: Dr. Angela Wu
Office Location: 104 McKinley Hall
Telephone No(s): 202-885-1476
e-mail Address: awu@american.edu
(Put CSC 520 on the subject line)

Textbook(s)/Reading:

Mark Weiss, *Data Structures and Algorithm Analysis in Java*, Addison Wesley, 2007.

Other References:

Cormen, Leiserson, Rivest and Stein, *Introduction to algorithms,* MIT Press
Michael T. Goodrich and Roberto Tamassia, *Data Structures and Algorithms*
Michael T. Goodrich and Roberto Tamassia, Algorithm Design

Course Description:

This course discusses the basic techniques and issues in the design and analysis of efficient algorithms, and, implementation and manipulation of data structures. Topics to be studied include linked lists, stacks, queues, trees, graphs, storage management, hashing, internal and external searching and sorting. Applications to various areas of computer science will be discussed.

Course Objectives:

Students learn to evaluate various data structures and algorithms, design appropriate data structures and efficient algorithms to manipulate the data structures, and to properly implement their designs in a high level language.

Grading Criteria/Course Requirements:

Your course grade is calculated based on the following

Homework and Projects	55%
Midterm Exam	20%
Final Exam	25%

Grading scale:

100 A 93 A- 90 B+ 87 B 83 B- 80 C+ 77 C 70 C- 65 D 60 F

E-mails:

Put CSC 520 on the subject line, otherwise it might get sent to the spam folder.

I only check emails at most twice a day. Don't expect instantaneous replies.

Emails are for short communications. I will not debug programs or explain computing concepts by email for obvious reasons. You should come to my office hours for these.

Policies:

No make-up exams will be given and **no** extra credit projects will be accepted in addition to or in lieu of them. If you miss a midterm with valid reasons **and** the instructor is notified prior to the exam, that exam will not count towards your course grade; instead, the final exam will carry a greater percentage of your overall grade. You must take the final exam at the scheduled time.

Late homework will not be accepted. Some of the homework problems will appear in the exams.

Programming projects will **not** be accepted if it is more than two weeks late.

Student participation in class is encouraged. If for some reason you have to miss a class, please let me know as soon as possible. Student should try to read the material on the topics before class. After each class, it is a good idea to try to reconstruct and work out the details of the algorithms and data structures discussed in class.

Please do not put assignments in my mailbox or in my office. Assignments will only be accepted in the first 5 minutes of class.

Late homework will not be accepted. Homework will be graded as satisfactory or unsatisfactory. Some of the homework problems will appear in the exams.

Programming projects turned in late (even if you cannot attend class) will be penalized 10% for each class session the submission is late. Programming projects will **not** be accepted if it is more than two weeks late.

Programming assignments must be clearly and properly documented. Part of the programming project grade will be based on the clarity of code and good documentation.

Please **turn off** your cell phones and beepers in class.

Incomplete ('I') Policy:

The policy of the Computer Science Department, the College of Arts and Sciences, and the University is that the grade of Incomplete (I) is rarely given. Department approval for a grade of I is only granted in unusual, documented circumstances. A grade of I is not approved in instances where students were unable to complete the course work.

Academic Integrity/Plagiarism:

The *Academic Integrity Code* for the American University describes standards for academic conduct, rights and responsibilities of members of the academic community, and procedures for handling allegations of academic dishonesty. Academic dishonesty as defined by the *Code* includes, but is not limited to: plagiarism, inappropriate collaboration, dishonesty in examinations (in-class or take-home), dishonesty in papers, work done for one course and submitted to another, deliberate falsification of data, interference with other students' work, and copyright violations (including both document and software copyrights). Copies of the *Academic Integrity Code* are available from the Office of the University Registrar.

Plagiarism is defined as taking the language, ideas, or thoughts of another, and representing them as your own. If you use someone's ideas, cite them; if you use someone's words, clearly mark them as a quotation. Plagiarism includes using another's computer programs or pieces of a program. Consult one of the many "writer's guides" that are available in the library and bookstores for citation practices. All instances of plagiarism will be reported to the Dean of the College of Arts and Sciences for appropriate action.

Schedule

(Tentative and subject to change)

Date	Topics (**Tentative and subject to change**)	Readings
8/25	Introduction	
	Algorithm Analysis	2,1-2,4, 7.2,
9/1	Examples - Big O, maximum subsequence	2.4, 7.6
	Linked lists	3.1-3.2, 3.5
9/8	stacks, queues	3.5, 3.7
	Trees, Binary trees	4.1, 4.2
9/15	Binary trees, binary search trees	4.3
	More on binary trees	4.6
9/22	AVL trees, B trees	4.4, 4.7
	Searching and Hashing	5.1 – 5.6
9/29	Heaps	6.1-6.3
	priority queues, heapsort	6.4
10/6	Union and find, Graphs – def. & rep.	8.1-8.5, 9.1
	Midterm	
10/13	Graph Searches, Shortest Paths	9.2, 9.3
10/20	Minimal Spanning Trees	9.5
	Depth First Search Applications	9.5, 9.6
10/27	Greedy Method	10.1
	Divide and Conquer	10.2
11/3	Dynamic Programming	10.3
11/10	Backtracking Algorithms	10.5
	Introduction to NP-Completeness	9.7
11/17	More on NP-Completeness	9.7 + Notes
11/24	Thanksgiving	
12/1	Sorting Algorithms	7.1 - 7.9
	Review and Catch up	
12/13	***Final Exam***	