NJIT New Jersey's Science & Technology University

A. V. Gerbessiotis | CS 610-102
January 27, 2014 | Spring 2014
Course Syllabus: General Information | Handout 1

Intensive study of the fundamentals of data structures and algorithms. Presents the definitions, representations, processing algorithms for data structures, general design and analysis techniques for algorithms. Covers a broad variety of data structures, algorithms and their applications including linked lists, various tree organizations, hash tables, strings, storage allocation, algorithms for searching and sorting, and a selected collection of other algorithms.

1.1 Contact Information

Instructor:	Alex Gerbessiotis	E-mail:	`alexg+cs610@njit.edu`
Office:	GITC 4213, 4th floor	Tel:	(973)-596-3244
Office Hours:	Mon and Fri 4:00-5:30pm		
Office Hours:	Also by appointment Mon/Wed/Fri		
Assistant:	TBA on course web-page		
Class Hours:	Mon 6:00-9:05pm, Room KUPF 205		
Course Web Page:	`http://www.cs.njit.edu/~alexg/courses/cs610/index.html`		

Print Handout 1 from Web-page and compare the printout to this document! They must be identical.

1.2 Course Administration

Prerequisites CS 505 or CS 335, and completion of all bridge course requirements.

Textbook Algorithm Design: Foundations, analysis, and internet examples. M. T. Goodrich and R. Tamassia. Wiley, 2001, ISBN 0-471-38365-1. *We abbreviate it in class as GT.*

CourseWork: **3 exams (including the final)**. Programming-based homeworks.

Grading scheme: 1000 points = HW(200) + Ex1 (333) + Ex2 (137) + Ex3(333). If a student collects 200 or more homework/programming points, 200 points will be added to the total; 199 or fewer programming points will be discarded.

HW1-3 3 or more assignments will be handed out. Each one is worth 100 points. They can be done in C, C++, or Java and will be tested on `afsconnect1.njit.edu` or `afsconnect2.njit.edu`.

Practice PS Four comprehensive problem sets PS1-4 will be periodically posted along with their solutions. Exam 2 will be based on these problem sets; the other two exams might also borrow ideas from them.

Exams Dates in Course Calendar; all exams in classroom. Exam 2 is closed-everything. The other two exams are open-textbook only; you may bring a hardcopy of the textbook but you may not borrow one during the exam. Exam1 (midterm) is on **Mar 10**, 2hours, 333 points. Exam2 (quiz) is on **Apr 14**, 60min, 137 points. Exam3 (final) is on **May 12**, 2hours, 333 points.

Due Dates Programs **MUST be received by email before noon-time** the day they are due; 30 points subtracted every 24-hour period starting with Monday noon-time. We strongly recommend that you use an NJIT email account to email your code; if you do not use one, do not complain about potential loss of emails (or NJIT's blocking of them). Read Handout 2 for details and the homework itself.

2.1 Course Objectives and Outcomes

Objective 1 Understand and formulate the input-output relationship of computational problems, and formulate the requirements, data and operations of abstract data types (ADT). data structures.

Objective 2 Learn, Understand and be able to describe data structures that represent the mathemtical model underlying an ADT.

Objective 3 Learn how to describe, derive and determine, the asymptotic performance of algorithms and data structures.

Objective 4 Learn how fundamental algorithms and data-structures operate, and understand their characteristics. Be able to choose among a variety of similar ones based on problem/program specification and requirements.

Objective 5 Learn how to compose more complex algorithms using as building blocks the fundamental algorithms introduced in class.

Objective 6 Learn how to compose more complex algorithms using the algorithmic design techniques introduced in class.

Outcome 1 Be able to accurately specify the input/output relationship of computational problems, and describe correctly fundamental ADTs.

Outcome 2 Be able to accurately specify the operations of fundamental ADTs and the data structures underlying them; be able to correctly describe the performance of algorithms for the operations on these data structures.

Outcome 3 Be able to asymptotically compare functions using $o, O, \omega, \Omega, \Theta$.

Outcome 4 Be able to solve recurrences using the master, the iteration/recursion tree, and the substitution method.

Outcome 5 Become familiar with a variety of algorithms for sorting, selecting and searching data and their performance characteristics (eg, running time, stability, space usage) and be able to choose the best one under a variety of requirements.

Outcome 6 Be able to understand fundamental algorithms and data structures and their performance and be able to trace their operations for problems such as sorting, searching, selection, operations on numbers, polynomials and matrices, and graphs.

Outcome 7 Be able to understand fundamental algorithm design techniques and understand how to use them to solve algorithmic problems.

Outcome 8 Be able to use the fundamental algorithms introduced in class to design algorithms for more complex problems and analyze their performance.

Outcome 9 Be able to use the design techniques and algorithms introduced in class to design algorithms for more complex problems and analyze their performance.

3.1 Tentative Course Calendar

Spring 2014				
Week	Wed	Exams	HW or PS	Comments
W1	1/27		HW1out,PS1*	
W2	2/3			
W3	2/10		PS2*	
W4	2/17		HW2out	
W5	2/24		HW1in	
W6	3/3			
W7	3/10	**Exam1**		Midterm Exam
W-	3/17			Spring Break
W8	3/24		HW2in ,PS3*	
W9	3/31		HW3out	
W10	4/7			
W11	4/14	**Exam2**	PS4*	
W12	4/21			
W13	4/28		HW3in	
W14	5/5			
W15	5/12	**Exam3**		Final Exam

* Problem Sets (PS) with solutions are not for credit. The following describes a tentative list of topics that is intended to be covered in class with indicative chapter pointers to GT. The lecture summaries contain detailed correspondence to chapters of the textbook.

Topics to be covered

```
T1 :Ch1,5.2     : Introduction (insertion-sort, fibonacci sequences),
                 Algorithm Analysis (Asymptotic Growth of functions, recurrences)
T2 :Ch1,4.1,5.2 : Algorithm Design Techniques (Incremental, Divide-and-Conquer)
                 Sorting(selection-sort,bubble-sort,merge-sort).
T3 :Ch2.1-2.3   : Elementary Data Structures and Trees. Tree traverals.
   Ch4.2         Union-find operations.
T4 :Ch2.4,9.3   : Heaps and Priority Queues. Greedy Algorithms and Huffman coding.
T5 :Ch4         : Quick sort. Complexity of sorting. Sorting in linear time
                 (radix-sort, bucket/count-sort). Selection.
T5 :Ch2.5-2.7   : Hashing.
T6 :Ch3         : Binary Search Trees and Balanced Binary Search trees.
                  m-way trees, 2-3-4 trees, B-trees.
T7 :Ch5         : Integer operations (addition and multiplication).
                 Matrix operations (addition and multiplication). Strassen's method.
                 Dynamic Programming and chained matrix multiplication.
T8 :Ch6         : Graphs and their representation. Graph traversals (DFS,BFS).
                 Strongly connected components. Topological sorting.
T9 :Ch7         : Weighted graph problems. Shortest-path problems (Dijkstra's).
                 All-pairs shortest paths and transitive closure (Floyd-Warshall).
                 Spanning trees (Prim's and Kruskal's algorithms).
T10:Ch9         :String and Pattern matching algorithms.
T11:Ch10        :Fundamental algorithms involving numbers. RSA.FFT.
T12:Ch13        :P and NP. NP-completeness.
```

Any modifications or deviations from these dates, will be done in consultation with the attending students and will be posted on the course Web-page. It is imperative that students check the Course Web-page regularly and frequently.

MISSING CLASS If you miss a class and there is no EXAM or PA due it's up to you to make up for lost time.

MISSING HW There are three scheduled programming assignments and not all are needed to collect the 200 points for the course. Plan ahead of time and submit early. No extensions are granted for any reason, medical or otherwise; only a snow cancellation may delay things.

MISSING EXAM If you miss EXAM2, then EXAM 3 will be used to determine a grade. No make-up will be given. For the other two exams valid documentation must be forwarded to the DEAN OF STUDENTS within 3 working days from the day the reason for the absence is lifted. The maximum accommodation will be the number of missing days to the exam date.

Programs Code must be ANSI compliant and neither hardware-specific nor OS-specific. Check Handout 2 for more information on the programming assignments. Programming problems are graded based on test instances decided by the grader on a test platform of the grader's choice (e.g. AFS). Do not expect partial credit if your code fails to run on all test instances unless you accompany your code submission with a detailed bug report.

Grading Written work will be graded for conciseness and correctness. Use formal arguments. Be brief and to the point and write clearly. Material covered in class and appearing in the relevant notes and chapters of the designated textbook can be used without proof. Excess programming points may be used to boost your course grade (we account them separately). DO NOT USE pencils to write down your solutions; if you decide to use a pencil do not complain later about grading.

Grades Check the marks in a written work and report errors promptly. **Make sure you report such problems to the grader or the instructor within two weeks from receipt** but **no later than the Reading Day**. If you believe a grade you received for the solution of a problem is not representative of your effort talk to the grader first and then to the instructor (if different). The final grade is decided based on the 0 to 1000 point performance with an adjustment made based on programming assignment performance. A student who collects at least 500 points and completes the minimum programming requirements should expect a passing grade (C or better). 900 points or more are usually required for an A including robust programming work.

Incomplete A grade of I(incomplete) is given in rare cases where work cannot be completed during the semester due to documented long-term illness or absence (e.g. unexpected national guard duty). A student needs to be in good standing (i.e. passing the course before the absence and satisfying programming requirements) and receives an I if there is no time to makeup for the documented lost time. Not showing up in the final will probably get you an F rather than an I.

Collaboration **Collaboration of any kind is NOT allowed in the in-class exams. Students who turn in code obtained through the Internet or otherwise, or is product of another person's/student's work, risk severe punishment, as outlined by the University. The work you submit must be the result of your own mental effort. In addition to University penalties, when a violation is detected and confirmed, the course grade will be reduced by one level (a C+ would become a C).**

Mobile Devices Mobile phones/devices and/or laptops/notebooks MUST BE SWITCHED OFF before the class exams. Switch off noisy devices (eg mobile phones) before you enter the classroom for a lecture.

Email/SPAM Send email from an NJIT email address. NJIT spam filters or us will filter other email address origins. Do not send course email to the instructor's email address unless there is a good reason (eg. you don't want the grader to read the email). Include `CS 610` in the subject line then. ■.

The NJIT Honor Code will be upheld; any violations will be brought to the immediate attention of the Dean of Students. Read this handout carefully!