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:	Wed 4:00-5:30pm and Thu 4:00-5:30	Else, by	appointment Mon/Thu
Assistant:	TBA on course web-page		
Class Hours:	Wed 6:00-9:05pm, Culm LEC1		

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. (The 335 is a relic from the past; the 505 is a not very active course either.)
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 project. Potentially one more unannounced quiz (To be determined)?
Grading scheme:	1000 points = PP(134) + Ex1 (200) + Ex2 (333) + Ex3(333).
PP	A programming project (PP) with 2 options each one worth 134 points. A student may submit one or two options but both options must be submitted in one shot; a grade of 0-40 points for an option is accounted as 0 towards the final grade. No more than 134 points of credit from the sum of the grades of the two options. See Handout 2 for more details.
Practice PS	Four comprehensive problem sets PS1-4 will be periodically posted along with their solutions. Exams may be based on these problem sets.
Exams	Dates in Course Calendar; all exams in classroom. Exams are open-textbook only; you may bring a hardcopy of the textbook but you may not borrow one during the exam. Exam1 (quiz) is on **Feb 10**, 60mins, 200 points. Exam2 (midterm) is on **Mar 2**, 2hours 333 points. Exam3 (final) is on **May 11**, 2hours, 333 points. A second quiz might be given around Apr 13; in that case Exam1's grade gets halved and this quiz will count for the cut 100 points. Or if there is a snowstorm and class cancellation on Feb 10, Exam1 may get rescheduled on **Apr 13**.
Due Dates	PP submitted **only by email and MUST be received BEFORE NOON-TIME**; 40 points subtracted from every option (after work is graded) at deadline plus 1 minute, and every 24-hour period thereafter. 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. Programs can be done in C, C++, or Java and will be compiled and tested on `afsconnect1.njit.edu` or `afsconnect2.njit.edu`.

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 mathematical 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 methods.

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 2016				
Week	Wed	Exams	PP or PS	Comments
W01	1/20		PPout, PS1*	
W02	1/27			
W03	2/03		PS2*	
W04	2/10	**Exam1**		
W05	2/17			
W06	2/24			
W07	3/02	**Exam2**		**aka Midterm**
W08	3/09			
W–	3/16			Spring Break
W09	3/23		PS3*	
W10	3/30			Mar28 is Last Day to Withdraw
W11	4/06		PS4*	
W12	4/13			Pop-up quiz maybe?
W13	4/20		PPin	
W14	4/27			
W–	5/04		No-class	**Reading Day**
W15	5/11	**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 If you miss a class, it's up to you to make up for lost time. If you miss Exam1 and the DEAN OF STUDENT SERVICES certifies your absence after receiving valid documentation from you, then the scaled Exam2 grade will be used. You MUST CONTACT DOSS within 2 working days from the day the reason for the absence is lifted. If you miss the unannounced pop-up quiz and you have a valid (DOSS granted) excuse for a make-up, Exam1 stays as is (and no make-up exam will be given).

Program Check Handout 2. Each program option is evaluated/graded first. Then the lateness penalty, if any, is applied. If the resulting grade for an option is 0-40 then the grade for that option is accounted as 0. The sum of the grades for the two options cutoff at 134 points is your programming grade.

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 or citation. Programming points discarded may be used, at the discretion of the instructor, to remove a student from a fail zone. This assumes no violation of the Collaboration policy below. **DO NOT USE pencils to write down your solutions; IF YOU DO USE PENCILS 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/return** and **no later than the (first) Reading Day** whichever comes first. First resolve issues with the grader, then the instructor (if different). The final grade is decided on a 0 to 1000 point scale. A student who collects at least 500 points (including some minimal programming work) should expect a passing grade (C or better). 900 points or more are usually needed for an A including robust programming work. (Assuming always no violation of the Collaboration policy.)

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 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 PROHIBITED in the in-class exams or the project. A student must turn in code that has fully been written by him/her. Any submitted code (even few lines) obtained through the Internet or otherwise, or is product of someone else's work or is common with another student submission, in the same or other section/course, risks severe punishment, as outlined by the University; all parties of such interaction receive automatically 0 and grade is lowered by one or two levels. The work you submit must be the result of your own mental effort and you must safeguard it from other parties; if you can't protect your home computer, use an AFS machine.**

Mobile Devices Mobile devices etc **must be switched off/shut** before and during class exams or lectures.

Email/SPAM Use an NJIT email address or your email might not reach us. Send email to the designated course email address and always include `cs610` in the subject line. ■.

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!