EECS 214 (also listed as 395/495): Data Structures and Data Management
Course Policies and Syllabus

Class Schedule:  MW 12:30-1:50 in Tech M-345

Instructor:  Aleck Johnsen, aleck [dot] johnsen [at] gmail [dot] com
Office Hours:  Monday 3-4:30pm, Ford 3-333, or by appointment

Course Description / Objectives:  This course studies the design, implementation, and analysis of abstract data types, data structures and their algorithms. The correct choice of data structures is important to enable efficient algorithms, especially when there are tradeoffs between initializing, accessing and manipulating data. Motivation drives design. Further, with the right data structures, the process of solving abstract problems can usually be made inherently visual; and correspondingly, can be learned on a very intuitive level.

Course Website:  NU Blackboard: https://courses.northwestern.edu
Information will also be available on my webpage: http://users.eecs.northwestern.edu/~acj861/

Required Text:  Introduction to Algorithms, Third Edition; Cormen, Leiserson, Rivest, Stein
(This book is optional, it is not necessary; programming projects can be completed using references available via natural Google searches as necessary.)

Another Resource:  Check out MIT OpenCourseWare: Introduction to Computer Science and Programming 6.00. (You will have to download videos to a player like iTunes.) The lectures teach the Python language.

Office Hours:  Wednesday 2-3:30pm, WilkinsonLab in Tech M-338
Office Hours:  Thursday 3:30-5pm, Wilkinson Lab in Tech M-338

TA Weekly Problem Session:  TBD

Kaitlyn MacIntyre, kmac2017 [at] u [dot] northwestern [dot] edu
Office Hours:  Thursday 5-8pm, Wilkinson Lab in Tech M-338

Office Hours Note:  If another class ever blocks us from Wilkinson, move to: Ford 3rd Floor, top of stairs

Prerequisites:  EECS 211 (Fundamentals of Computer Programming II) or EECS 231 (Advanced Programming for Computer Engineers); it is recommended that students have completed or are currently taking EECS 310 (Mathematical Foundations of CS)

Grading:
5 Written Homeworks:  25% total as 5% each
3 Programming Projects:  25% total as 5%, 10%, 10%
Micro-quizzes and Program-Zero:  5% total
Midterm Exam:  15%
Final Exam:  30%

Grading Notes:
- Program-Zero, worth 1%, will require installing Python 2.7 or 3.4 in the first ~10 days and submitting a very simple program.
- 8 Micro-quizzes, each worth 0.5%, will be take home “10-minute concept checkers” via Blackboard. The objective is to help students identify any gaps in understanding of lecture material before they become serious setbacks on homework, programming projects, or exams.
- Late homeworks/programs will earn 60% credit until solutions are posted, at which point they can no longer be accepted for credit. Homeworks are due in class or in my office up to 2pm on the due date (10 min. after class). Programming assignments are due by midnight. Deadlines are strict.
- Final Exam will be cumulative with more weight on second-half material.
- 2% subjective extra credit available for participation in class.
- There will not be any curving of grades.
- **Collaboration Policy:** Students are encouraged to theoretically solve both written and programming projects together. However, all homework should later be written up / coded and turned in independently as individuals. The course will operate under a "one person's-worth of work deserves one person's-worth of credit" policy. If you want to turn in a single assignment for multiple people, that's fine– put all names on it. But the credit will be split amongst the group. For example, if 3 names are on a homework assignment scoring 90%, each student will earn only 30%. Obviously this is not recommended. Further, if during grading it is discovered that submissions are identical as a result of copying, then not only will the credit be split, but an additional 50% penalty will be applied to all.

**Programming Project Notes:**
- This course will use Python. We will not do anything too complicated, so versions 2.7 or 3.4 will both be sufficient. Instructions to download Python for Windows and Mac will be posted to Blackboard.
- Programming projects will be assigned early with long deadlines. It is possible that not all material needed to finish them will have been covered when first assigned. Students are expected to budget their own time. If you are having problems with the language, ask for help early.
- Projects will provide a starting template outlining required classes, inheritances, methods, etc.
- Programs will be graded by test suites. To accommodate this, projects will be turned in as a single file (filling in the template) including all data structures and methods. These objects should be edited to end with "." and your student ID. For example, for a class "Node" I would edit it to be a class named "Node_acj123". If you write any test code, it needs to appear within a "__main__" block.
- This is a course in fundamental data structures. Programming projects are not intended to test whether or not you know how to use built-in structures like lists and dictionaries. Rather you should understand how to build these. Projects will specifically preclude the use of certain built-in types. To enforce this, submissions might be checked by text-searches and/or manual spot checks of the code.

**Tentative Syllabus:**
- Wed 9/24: Introduction to Data Structures
- Mon 9/29: Run-time analysis, Recurrences
- Wed 10/1: Array Lists, Linked Lists, Dynamic Re-sizing
- Mon 10/6: Abstract Data Types, Stacks, Queues, **Program-Zero Due**
- Wed 10/8: Trees, Traversals, **HW 1 Due**
- Mon 10/13: Binary Search Trees
- Wed 10/15: Red / Black Trees, **HW 2 Due**
- Mon 10/20: Hash Tables, Hash Functions, **Program 1 Due**
- Wed 10/22: Collision Handling, Probability
- Mon 10/27: Skip Lists
- Wed 10/29: **Midterm Exam**
- Mon 11/3: Graphs, Graph Representations, BFS, DFS
- Wed 11/5: DFS, Graph Algorithms: Topological Sort, **HW 3 Due**
- Mon 11/10: Shortest Paths
- Wed 11/12: Minimum Spanning Trees, **Program 2 Due**
- Mon 11/17: Priority Queues, Heaps
- Wed 11/19: Priority Queues, Heaps, **HW 4 Due**
- Mon 11/24: Sorting, Divide and Conquer, Merge Sort
- Wed 11/26: Merge Sort, Heap Sort
- Mon 12/1: Topics: Amortized Cost, **HW 5 Due**
- Wed 12/3: Topics: Maximum Graph Flow, **Program 3 Due**

**Friday 12/12, 3-5 pm:** Final Exam