MARYLAND APPLIED GRADUATE ENGINEERING

ENPM809X
Spring 2019

Course Syllabus

Special Topics in Engineering; Data and Algorithms

Learning Outcomes

The course will teach the algorithms, heuristics, and code development skills in solving numerous sorting, graph theory, string processing, and job scheduling problems. It provides both a broad coverage of basic algorithms and data structures and in-depth coverage on selected topics. Moderate to heavy programming in C is expected. Through this study the students will improve their problem solving skills via various algorithm development examples and exercises.

Required Resources

Course website: **elms.umd.edu**

Textbook (mandatory):

Introduction to Algorithms
T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein
Third Edition
MIT Press.
ISBN: 978-0-262-53305-8

Several assignments will be directly from the textbook.

Dr. Emre Gunduzhan
egunduzh@umd.edu
egunduzhan@hotmail.com

Class Meets
Thu 4:00pm – 6:40pm
Online

Office Hours
TBA

Prerequisites
Basic C Programming

Course Communication
Important reminders and course information will be via ELMS Announcements

Use ELMS Discussions for any questions or comments about course material

Feel free to e-mail for general questions and everything else

Campus Policies

It is our shared responsibility to know and abide by the University of Maryland's policies that relate to all courses, which include topics like:

- Academic integrity
- Student and instructor conduct
- Accessibility and accommodations
- Attendance and excused absences
- Grades and appeals
- Copyright and intellectual property

Please also visit the **Graduate Policies and Procedures for Review of Alleged Arbitrary and Capricious Grading** and follow up with me if you have questions.

Activities, Learning Assessments, & Expectations for Students

There are 14 lectures in the course. I expect students to attend the lectures so that they can actively participate in discussions. Each lecture will start with a review of the previous topics and I expect the students come to the class prepared to answer questions. Although attendance is not mandatory, participation to in-class and ELMS discussions constitutes part of the overall grade.

There will be homework assignments in the form of questions or small programming tasks every two weeks throughout the semester. The homework solutions should be turned in electronically using ELMS.

There will be two projects, involving some programming. Each project will take 3-4 weeks to complete. The projects should also be turned in electronically using ELMS.

There will be a final examination.

The overall grade will be based on participation to discussions, homeworks, projects, and the final exam.

All homeworks, programming projects and the final exam are individual assignments. Academic dishonesty, including cheating, fabrication, plagiarism, copying solutions or code from another source (such as another student, internet, etc) will not be tolerated.

Course-Specific Policies

For this course, some of your assignments will be collected via Turnitin on our course ELMS page. I have chosen to use this tool because it can help you improve your scholarly writing and help me verify the integrity of student work. For information about Turnitin, how it works, and the feedback reports you may have access to, visit Turnitin Originality Checker for Students

Grades

All assessment scores will be posted on the course ELMS page. If you would like to review any of your grades (including the exams), or have questions about how something was scored, please email me to schedule a time for us to talk.

Late work will not be accepted for course credit so please plan to have it submitted well before the scheduled deadline. I am happy to discuss any of your grades with you, and if we have made a mistake I will immediately correct it. Any formal grade disputes must be submitted in writing and within one week of receiving the grade.

Assessments	Number	Category Weight
Participation to Discussions	5	10%
Homeworks	5	30%
Projects	2	30%
Final Exam	1	30%

Each assessment will be scored out of a maximum of 100 points. The average of the individual assessment scores in a category will form the score for that category (obviously no averaging is needed for the final exam). The total score will then be calculated by a weighted average of the category scores using the weights shown in the above table.

Final letter grades are assigned based on the total score earned. To be fair to everyone there are clear standards and they are applied consistently, so please understand that being close to a cutoff is not the same as making the cut ($89.99 \neq 90.00$). It would be unethical to make exceptions for some and not others.

Final Grade Cutoffs									
		+	85	+	65	+	50		
A	>=95	B	78	C	60	D	45	F	< 40
-	90	-	70	-	55	-	40		

Course Schedule

Jan 28	Introduction to algorithms and data structures; Review of C programming	
Feb 04	Growth of functions; Divide and conquer algorithms; Insertion sort and merge sort	
Feb 11	Randomized algorithms; Heaps and heapsort	HW1 due
Feb 18	Priority queues; Quicksort; Sorting in linear time	
Feb 25	Elementary data structures; Hash tables	HW2 due
Mar 04	Rooted trees; Binary trees; Red-black trees	
Mar 11	Dynamic programming	
Mar 18	***** Spring Break – No Class *****	
Mar 25	Greedy algorithms; B-trees	Project1 due
Apr 01	Advanced data structures	
Apr 08	Graphs and elementary graph algorithms	HW3 due
Apr 15	Graphs (cont'd); Shortest paths	
Apr 22	Shortest paths (cont'd)	HW4 due
Apr 29	Maximum flow	Project2 due
May 06	Multithreaded algorithms	HW5 due
May 13	**Final Exam**	