UMSL

# Syllabus: CmpSci 5130

## *Advanced Data Structures and Algorithms*

## Department of Mathematics and Computer Science

### Semester

*SP/SS/FS XXXX*

# Instructor Details

**Name:** *Put name*
**Office Hours:** *Put hours, 1.5 hrs/wk per course for FT faculty, 3 hours a week for adjunct*
**Office Location:** *Put location, can be in office, in class, online, etc.*

### Submission and Communication

*Your specific policies and procedures regarding submissions, late submissions, communication means, etc.*

### Scoring

*List any additions/changes you want to make to the course details below, such as using quizzes, attendance requirements, etc. Keep in mind you cannot change the course details, you can only work within what it says. So if you want to add quizzes and they are not listed, scoring for quizzes has to be added to homework, tests, or another part. If the course detail gives ranges for grading, you have to provide specific values within these ranges.*

### Incremental Grading

*Provide information if used.*

### Schedule

*If the course detail does not state detailed timing or sections/topics, you may put them here. Keep in mind you cannot remove topics and if the course is coordinated you may have to follow topic allocations.*

# Course Details

### General Policies

Lecture recordings, audio or video, are not permitted unless the instructor explicitly allows it.

We follow the university policies regarding excused EX and EX-F drops.

Students are given and are expected to sustain a positive learning environment in class. This means positive conduct in class, no late walk-ins or early walk outs without a good explanation or a prior arrangement, and if on-line access is available in class - not using it for anything that isn't class related. Students not meeting these standards may be asked to leave the classroom.

All in and out of class work for grade should be done independently (except for group projects). Homework can be discussed with others, but the final work (code, answer, etc.) must be independent. Programs may be discussed up to design, but no code is allowed to be shared except for what is presented in class. Help can always be sought and received. However, help for assignments should be generic on the subject matter or very narrowly focused on specific problem not being the central point in the assignment.

## Course Description

Prerequisites: Graduate standing in Computer Science. This course covers the design of efficient data structures and algorithms, as well as an advanced analysis of the time and space complexities of iterative and recursive algorithms. Student will learn a variety of techniques including dynamic programming, greedy algorithms, various graph algorithms, and NP-completeness and approximation algorithms.

## Text and Other Materials

Course textbook: *Introduction to Algorithms* by T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein; 3rd edition; ISBN 978-0-262-03384-8 or ISBN 978-0-262-53305-8.

## Course Schedule

*Schedule is tentative and subject to change.*

| WeeK | Chapter | topic |
|---|---|---|
| *1* | 1, 2, 3 | Refresher on complexity and divide-and-conquer |
| *2* | 15 | Dynamic programming |
| *3* | 15, 16 | Dynamic programming and greedy algorithms |
| *4* | 16 | Greedy algorithms |
| *5* | 17 | Amortized analysis |
| *6* | | Review and **Exam 1** |
| *7* | 21 | Data structures for disjoint sets |
| *8* | 22 | Elementary graph algorithms |
| *9* | 23 | Minimum spanning trees |
| *10* | 23, 24 | Minimum spanning trees and single-source shortest paths |
| *11* | 24, 26 | Single-source shortest paths and maximum flow |
| *12* | 26 | Maximum flow |
| *13* | | Review and **Exam 2** |
| *14* | 34 | NP-completeness |
| *15* | 35 | Approximation algorithms |
| *16* | | **Final Exam** |

## Course Objectives and Learning Outcome

Upon completion of this course, students should be able to:

- Analyze space and time complexities of algorithms
- Improve complexity analysis using amortized methods when applicable

- Design efficient data structures and recognize relationships between these structures and the algorithms that operate on them
- Design divide-and-conquer, dynamic programming, and greedy algorithms and determine which of these approaches is best suited for a given task
- Understand and be able to apply several graph algorithms to various network problems
- Model real-world problems as linear programs and determine suitable algorithms for solving
- Understand the basics of NP-completeness, including its implications for computations

### Course Grading

We will use the standard 10% grading scale: 90% and above for an A, 80% and above for a B, 70% and above for a C, 60% and above for a D, else F. All grades throughout the course will be posted on Canvas.

| | |
|---|---|
| *Homeworks, quizzes, etc.* | 15% |
| *Projects* | 25% |
| *Exams and comprehensive final* | 60% |

# University Policies and Information

http://www.umsl.edu/~webdev/mathematics/files/pdfs/cs_umsl_syllabus_university.pdf