ECS60: Data Structures and Programming Syllabus *

Instructor: Dr. Rob Gysel

Winter 2018

E-mail: `rsgysel at ucdavis dot edu`
Office Hours: See Piazza->Resources->Staff
Office: 3023 Kemper Hall
Discussion Sections: A01: W 4:10-5pm Olson 250
A02: T 9-9:50am Veihmeyer 212
A03: F 2:10-3pm Art 204

Lecture:
MWF 9-9:50am
Scrub Oak 160

Classrooms are subject to change. Refer to ScheduleBuilder for the most recent information.

Class Roadmap

I utilize a number of online resources for the course. Over the years I've tried to narrow the number of sites down. This has been unsuccessful; for example, I previously eliminated Piazza in favor of Canvas but students unanimously wanted Piazza's Q&A functionality over Canvas. Use the following reference to navigate the course.

Program Submission: Submit to *Kodethon*

Written Homework: Submit to *Gradescope*

Exams: View and request regrades of in-class exams on *Gradescope*

Grades: *Canvas*

Homework Files: *Canvas*

Q&A Forums: *Piazza*

To join these sites:

Gradescope: Create an account at `gradescope.com` and use entry code MP58X2. **When creating your account, carefully enter your student ID number. If it differs from your actual student ID number, any grades from Gradescope will not transfer to Canvas and you will receive a 0 for those grades.**

Piazza: Create a piazza account, visit `piazza.com/uc_davis/winter2018/ecs60` and use access code `datastructures`.

*Last updated 01/08/2018

Kodethon: Log in with your ucdavis e-mail account at `kodethon.com` Search for course *Winter 2017: ECS60* and use password `datastructures`.

zyBooks: Create an account at `learn.zybooks.com` and subscribe using code `UCDAVISECS60DataGyselSpring2018`.

Course Description

Homeworks, lecture notes, and updated lecture schedules will be posted on Canvas as the quarter progresses. There will be approximately four written homeworks, four programs, two midterms, and one final exam. Written homeworks will be submitted to Gradescope and programs will be submitted to Kodethon. Midterms and homework quizzes are graded and returned via Gradescope.

We will use an electronic textbook from zyBooks for the course. Reading it is worth 5% of your grade, and due dates for the reading are found on the zyBook. Generally, the reading for each lecture is due at 11:59pm the day *before* that lecture takes place. This is to get you thinking about the material before you come to class so that our lecture time is more fruitful. Please do not stress abut potentially losing "reading points" by having reading due prior to lecture; the losses you take here will be made up when you get to the homework, projects, and exams, all of which are worth a much greater percentage of your grade.

You may work in groups of two for the programming projects; however, I strongly encourage you to work alone. If you choose to work in a pair, **you are not permitted to have the same partner for more than two projects**. It is more important to improve your programming skills and be ready for the workforce than it is to get a slightly better letter grade in the course[1] Students who work alone and receive a B- or better for *all* programming projects will receive 1% extra credit at the end of the quarter.

Course Objectives

1. Learn theoretical fundamentals of sorting algorithms, data structures, elementary graph algorithms, and complexity analysis (Big-O).
2. Apply your OOP programming knowledge to implement data structures. You may use C++, Java, or Rust to implement your programs for this course.
3. Develop critical thinking & mathematical reasoning skills.

Required Readings

Required: *University of California, Davis ECS 60 Gysel Fall 2017: Data Structures Essentials*, by Roman Lysecky et. al. (zyBooks). zyBook Code: UCDAVISECS60DataGyselFall2017 (Don't worry about the code: this book is for winter quarter, but it was acquired in Fall)

[1]When I mention that focusing on grades is less important, the usual response is "well sure, except I'm trying to get into grad school." Yes, grades matter for graduate school. But if you are artificially inflating your grade, you risk attending a graduate school that is beyond your ability.

Optional: You may purchase and complete one of the zyBooks for up to 3% extra credit. The intent is for you to learn an OOP language different than one you already know. Many instructors use C++ here, so if you have no C++ knowledge, I strongly recommend you pick the C++ book. On the other hand, if you already have C++ knowledge, it is in your interest to learn Java since you will use it in ECS140 and it is a requirement of a large number of programming jobs. You may choose to learn a new language while writing programs for the course in a language you are familiar with. For example, if you want to learn C++ but write your programs in Rust, you are welcome to do so. If you wish to switch to another language from one project to the next, you are welcome to do so.

Grading

Programs 25%, Written Homework 10%, zyBooks 5%, 2 Midterms 30%, Final Exam 30%. Letter grades will be approximately: A = 90+%, B = 80-89%, C = 70-79%, D = 60-69%, F < 60%. If necessary, I will curve exams to a mean of 78 and standard deviation of 10. Scores that are entered into Canvas are your final, curved scores. The letter grade that appears in Canvas is your current grade in the course, with the exception of A+'s, which will be given to the top 3 students in the class. Occasionally, your Canvas grade is lower than what is displayed. This may occur if you have not turned in assignments or if the zyBooks reading scores have not been updated on Canvas[2]. **When in doubt, calculate your grade using the above percentages.**

Homework Policies

Homework assignments must be typed in LaTeX[3] and submitted to Gradescope as a PDF[4]. These homeworks are not graded, but serve as a basis for the weekly in-lecture quizzes (closed notes). I expect you to be able to pass the quizzes based on the knowledge you derive from completing the homeworks.

Your homework workflow will look something like this:

1. Obtain the homework assignment from Canvas.
2. Solve the homework, format solutions via LaTeX, and submit the pdf to Gradescope by the due date.
3. After the homework is due and before the in-class quiz, review solutions posted to Canvas.
4. An in-class quiz occurs at the end of lecture. These are typically 10-15 minutes long, with 2-3 short question on one side of one page.

[2]We must transfer scores from zyBooks to Canvas manually, so we tend to do it in batches

[3]FAQ: Why are you making us typeset homework in LaTeX? This takes forever. A: Two reasons: first, spending a significant amount of time on write-ups is a fact of adult life; second, the policy is to encourage concise, clear answers to homework questions. In the interest of saving time, you may hand-draw your figures and insert the image into your LaTeX file using the `includegraphics` command.

[4]FAQs: Q1: Can I use an Word Processor such as MS Word to typeset the homework? A: No. Q2: Do I need to learn all of LaTeX to do the homework? A: No, and I discourage this. Use the template file on Canvas and modify it as needed.

I encourage you to discuss the material with fellow students, TAs, and myself. The goal of this policy is to free you from worrying about what sources you may use for your homework. That said, all written assignments must be your own work, and you should be able to explain why you believe your solution is correct if asked to do so. If it becomes clear that you do not understand your submitted pdf, we reserve the right to subtract points from your homework score, regardless of its correctness.

Submitting carbon copies of online sources or another student's submission will result in a referral to student judicial affairs. *Piazza is available as an online Q&A platform.* An important part of your education is to learn to ask and address questions in a respectful and effective manner.

Exam Policies

The following behavior during exams is not acceptable in this course, and each is grounds for a referral to student judicial affairs:

- Talking to other students.
- Looking at other student's papers.
- Wandering eyes, e.g., looking at the ceiling to think.
- Having electronics (including smart watches) out of your backpack.
- Wearing hats with forward-facing bills. Take your hat off or turn it around.

Other policies appear on the front page of each exam. For example, it is not acceptable to sit next to your study partners for exams. Cheating wastes your time and my time, your tuition money, university resources, and devalues the degrees conferred by our institution. Don't do it. **Your degree is only as meaningful as the knowledge you have acquired to get it.**

Important Dates

First class: Monday 1/8

Holiday: Martin Luther King Jr. Monday 1/15/17

Midterm 1: Wednesday 1/24/17

Holiday: President's Day Monday 2/19/17

Midterm 2: Wednesday 2/21/17

Last class: Friday 3/16

Final Exam: Tuesday 3/20, 10:30am-12:30pm

Weekly Schedule

Please refer to zyBooks for zyBooks reading due dates.

Week 01, 01/08 - 01/12:

Mon. Lecture: Introduction, Binary Search, Algorithm Analysis
Reading: *zyBooks* Ch. 1.1-1.4 or 8.1-8.4 (zyBooks assigns both due, but you only need to do one or the other)

Wed. Lecture: Binary Trees, $\log(n)$ and 2^n Timebounds
Reading: *zyBooks* Ch. 4.1 *Canvas (supplemental)* Trees.pdf

Fri. Lecture: *(Sorting Algorithms)* Selection Sort, Sorting Lower Bound, Insertion Sort
Reading: *zyBooks* Ch. 1.5-1.7 or 8.5-8.7 (zyBooks assigns both due, but you only need to do one or the other)

Week 02, 01/15 - 01/19:

Mon. No class: Martin Luther King Jr. Holiday

Wed. Lecture: Quicksort, Mergesort
Reading: *zyBooks* Ch. 1.8-1.9 or 8.8-8.9

Fri. Lecture: *(Lists, Stacks, & Queues)* ADTs, Singly & Doubly-linked lists
Reading: *zyBooks* Ch. 2.1-2.7

Week 03, 01/22 - 01/26:

Mon. Lecture: List Traversal, Stacks, Queues
Reading: *zyBooks* Ch. 2.8-2.13

Wed. Midterm 1 (Comprehensive up to 1/19/17)

Fri. Lecture: *(Tree Data Structures)* Binary Search Trees (BST), BST Search / Insert / Remove
Reading: *zyBooks* Ch. 4.2-4.5

Week 04, 01/29 - 02/02:

Mon. Lecture: BST Traversal / Insertion Order
Reading: *zyBooks* Ch. 4.6-4.8

Wed. Lecture: Heaps, Binary Heaps, Building Heaps in Linear Time
Reading: *zyBooks* Ch. 4.9

Fri. Lecture: Priority Queues, Treaps
Reading: *zyBooks* Ch. 4.10

Week 05, 02/05 - 02/09:

Mon. Lecture: **(Balanced Trees)** AVL trees *zyBooks* Ch. 5.1-5.3

Wed. Lecture: Red-Black Trees (RBT), RBT Rotations *zyBooks* Ch. 5.4-5.5

Fri. Lecture: RB Insertion, Removal *zyBooks* Ch. 5.6-5.7

Week 06, 02/12 - 02/16:

Mon. Lecture: *(B-Trees)* B-trees (BT), BT Search / Insert
Reading: *zyBooks* Ch. 7.1-7.3

Wed. Lecture: BT Rotation / Removal
Reading: *zyBooks* Ch. 7.4-7.5

Fri. Lecture: *(Disjoint Sets)* Equivalence Relations & Partitions
Reading: *Canvas (supplemental)* Partitions.pdf

Week 07, 02/19 - 02/23:

Mon. No class: President's Day Holiday

Wed. Midterm 2 (Comprehensive up to 2/14/17)

Fri. Lecture: Disjoint-Set (aka Union-Find) Data Structure

Week 08, 02/26 - 03/02:

Mon. Lecture: Disjoint-Set Path Compression, Ackermann's Function

Wed. Lecture: *(Graph Algorithms)* Graphs, Graph Representations
Reading: *zyBooks* Ch. 6.1-6.3

Fri. Lecture: BFS, DFS
Reading: *zyBooks* 6.4-6.5

Week 09, 03/05 - 03/09:

Mon. Lecture: Connected Components, Topological Sort
Reading: *zyBooks* 6.6-6.7

Wed. Lecture: Dijkstra's Algorithm
Reading: *zyBooks* 6.8

Fri. Lecture: Bellman-Ford Algorithm
Reading: *zyBooks* 6.9

Week 10, 03/12 - 03/16:

Mon. Lecture: *(Hash Tables)* Hash Tables, Chaining
Reading: *zyBooks* 3.1-3.2

Wed. Lecture: Open Addressing
Reading: *zyBooks* Ch. 3.3-3.5

Fri. Lecture: Catch-up or TBA