Data Structures Course Calendar & Syllabus

COMP 271: Data Structures, Spring 2019 at Loyola University Chicago

Quick links to: Sakai, Trello board, jump to Syllabus section

All future topics are tentative. Anything in parenthesis is optional. Consult the Loyola Academic Calendar for relevant registration, add/drop, and withdrawal dates.
Due dates: If an assignment is spread over multiple days, for purposes of due dates, assume it is given last day of that block (due one week after that unless noted otherwise).

DATES	TOPIC	EXAMS AND QUIZZES	ASSIGNMENTS	READINGS AND TUTORIALS
Jan 14	Course Overview, Introductions	Short java quiz (in class, docx for reference), Java sakai quiz [sakai]	Precourse survey [Form] Java ad [Trello] "Interesting program" assignment [Trello]	Oracle's Java Tutorial [link] - skip Generics for now. (all optional: Bailey chapter 1, K&W: Appendix A.1-3, A.7-8 Chp 1.1-6)
Jan 16	Java Fundamentals [slides 1-46] Quantifying algorithms (Big-O) [slides]	Big O sakai quiz [sakai]	Big-O HW [doc]	Bailey 5.1 (more optional: Java programming "cheat sheet" [link], K&W 2.1-4, TIOBE index: [link] to answer "Why Java?")
Jan 21	NO CLASS - Martin Luther King Holiday			
Jan 23 Jan 28	Java fundamentals discussion. An assortment of sorts [slides]	Sorts sakai quiz [sakai]	Sorts HW [gDoc]	Bailey 6.1-6.5 (K&W 8.1-8.7, 8.9, sorts animation [link1] [link2])
Jan 30	Class canceled due to extreme weather			
Feb 4	Java collections: overview, List, Set, and Queue [slides, parts 1&2]	A previous class's exam for study prep [docx]	StackSet HW [gDoc] (due Feb 18)	Collections Tutorials and/or the API docs (Generics section of Oracle's Java Tutorial [link])

Feb 6	Java collections: Map, hashing [slides, part 3]	Java collections sakai quiz [sakai]	Collection Selection HW [gDoc] (due Feb 20)	For review: lecture slides, HW, quizzes, … (somewhat reluctantly, here's a big O cheat sheet [link])
Feb 11	EXAM I			
Feb 13	Presentation selection (in class)		Data structure short presentation selection [gDoc] [slides from past course]	
Feb 18	Queues (FIFO and Stacks) [gSlides]	Exam I handed back and discussed	Project "brainstorming" paragraph [Trello]	Bailey 10.1-10.2 (K&W 4.1-4.7)
Feb 20	LinkedList [gSlides]	LinkedList and Queue sakai quiz [sakai]	Prepare for presentations!	Bailey skim 9.4, read 9.5 (K&W 2.5-2.9)
Feb 25	Data Structure/ Algorithm Presentations [gDoc] [slide link in email]			
Feb 27	Project Selection (in class - please complete brainstorming assignment before this)			
Mar 4&6	NO CLASS - Spring Break			
Mar 11	Binary search trees [gSlides]			Bailey 12.4,12.6 (K&W 6.1-3, 6.5)
Mar 13	Priority Queues/ Heaps [gSlides] SkipList overview [slides] SkipList code review organization	Trees and Heaps sakai quiz [sakai].	Project proposal [gDoc] (discussed previously, but assigned today - due Mar 20) Organize for SkipList code review Mar 25-27 [gDoc]	Bailey 13.1, 13.3, 13.4.1-2 (K&W 6.6, and [heap demo]) JCF list interface specification.
Mar 18	SkipList code review preparation and project work			
Mar 20	Maps: hash tables [gSlides]	Maps sakai quiz [sakai]		Bailey 15.4 (K&W 7.1-7.5)

		Previous Exam II [gDoc] for study		
Mar 25	SkipList code reviews (20 minutes each)	Groups 1-3	SkipList code walkthroughs in class [gDoc]	Exam prep advice: slides > sakai quizzes > homeworks > readings
Mar 27		Groups 4-6,		
Apr 1	Exam II			
Apr 3	Graphs: intro [gSlides]	Final Project report [gDoc] and presentation [gDoc] assigned - due Apr 17 before class		Bailey 16.1-2 (K&W 10.1-2)
Apr 8	Graphs: implementations [gSlides]	Exam II passed back		Bailey 16.3 (K&W 10.3-4)
Apr 10	Graphs: advanced algorithms [gSlides]	Graphs sakai quiz [sakai]		Bailey 16.4 (K&W 10.5-6).
Apr 15	Final project presentation week	Group evals (done in class)	Final project report and slides DUE wednesday before class - no exceptions	Presentation order in assignment [gDoc]
Apr 17				
Apr 22	NO CLASS - Easter Break			
Apr 24	Final exam from previous semester for review [pdf] (interesting visuals for review [link]) Technical interview questions review activity [gDoc] Data structure design principles [gSlides] Last time for final exam preparation - BRING QUESTIONS! Parting thoughts			
Thurs, May 2	Thursday, 1:00-3:00pm Final Exam			

Syllabus for Data Structures

COMP 271: Data Structures, Spring 2019 at Loyola University Chicago

General Course Description

- **General course description (from the course catalogue):**
  This course introduces key data structures such as lists, sets, and maps, as well as their implementations. Performance and analysis of algorithms are covered along with applications in sorting and searching.

  This continuation of COMP 170: Introduction to Object-Oriented Programming introduces the concepts of data abstraction and data structure, including stacks, queues, lists, sets, and trees. The issues of implementing a data structure in a language such as Java are examined using classes, arrays, and linked structures. Sorting and searching techniques are analyzed. The concepts of correctness and efficiency of algorithms are developed. Time/space comparisons of iterative algorithms with recursive algorithms are made. The course includes several major programming projects. A weekly lab component is required.

  This course is programming intensive. Additional time working on reading, preparing, and programming will be required outside of class time.

- **Specific flavor of course**:
  - Our examples during lecture will be primarily from current java implementations of data structures
  - The course is roughly sectioned into 3 parts:
    - Theory, Concepts, Interfaces, Overview of Java Collections Framework (EXAM I).
    - Implementation of Data Structures (EXAM II)
    - Applications of advanced data structures (FINAL EXAM)
  - We will aim for final projects that are suitable for a portfolio (e.g. an android application making intensive use of a data structure implementation, a desktop app using a data structure to efficiently search protein or gene sequences...)
- **Prerequisites**
  - COMP 170: Introduction to Object Oriented Programming (or COMP 215)
  - COMP 163: Discrete Structure (pre or co-requisite).
  - Students are expected to be able to read, write, and debug basic computer programs using standard tools including compilers and editors. Students are expected to know the basics of Object Oriented Programming and be able to use

classes and methods they or others have written. See me if you have any questions or concerns about your preparation.

- Course mechanics: Most days of the course will have the following components.
    - **Discussion** of past readings, quizzes, or assignments review
    - **Today's Topic** with a brief lecture overview and expectations from assigned readings
    - Current **Assignments**
    - **Breakout** into individual or groups for help on assignments/readings

Course Details

- Time/location
    - MW 1:40-3:35pm, Lake Shore Campus, Crown Center 105
- Instructor and TA:
    - Instructor: Dr. Mark V. Albert
    - Contact: mva@cs.luc.edu (much preferred).
    - Discussion times: best before class any day or after class, by appointment if it will take a significant amount of time.
        - also available on google hangouts or skype under the name markvalbert, video available.
        - Office: Doyle Center 309
        - Instructor office hours: Mondays 12:20 - 1:20 (before class monday)
    - Teaching Assistant: Jess Conway, jconway6 @ luc.edu
        - He will be responsible for organizing the Trello board where assignments are to be posted, along with feedback.
        - TA Office Hours: Fridays 2:30-5pm Doyle Hall garden level
- Tutoring availability:
    - We have tutors that are all qualified to assist with COMP 271. Hours are set early in the semester, so give some time for them to be solidified.
    - http://www.luc.edu/cs/schedules/tutoringhours/

- **Course texts:**
    - Data Structures in Java: for the principled programmer. Bailey. 3rd ed. http://www.cs.williams.edu/~bailey/JavaStructures
        - This textbook is free and online. The appropriate sections will be referenced in the calendar for each topic. Note, we will mention slight variations throughout the course as it uses an older version of Java.
    - (optional, but recommended) Data Structures: Abstraction and Design Using Java. Elliot B. Koffman and Paul A. T. Wolfgang. 3rd ed. Wiley

        - The online version is recommended. You can try the online text free for 14 days at **www.WileyStudentChoice.com**
        - Attempts will be made to make the lecture content, expectations for preparation, and assignments relatively independent of the text, however, it is recommended that you purchase the text this semester, as the additional provided materials may not be sufficient.
    - Problem Solving with Algorithms and Data Structures using Python
        - https://interactivepython.org/runestone/static/pythonds/index.html
        - A great free online resource for those that want to explore these topics in Python as well. We will not be covering aspects of Python in this class.
    - Other Java resources:
        - The official documentation for Java version used in class is from Oracle https://docs.oracle.com/javase/8/docs/api/
          It is useful to look up specific terms, concepts, and other things you want to see.
        - Oracle has a much material to help Java programmers: start from this link:
          http://www.oracle.com/technetwork/topics/newtojava/overview/index.html

- Software:
    - *The following selections are to help you become accustomed with many different programming strategies. Some assignments will specify which tools to use, while the majority, when unspecified will allow you to select what is best for you.*
    - **Command line/Scripting:** There is no simpler way to stitch various programs together, and all you need is a command line and text editor.
    - **IntelliJ:** This is an IDE (integrated development environment) that is an industry standard. You are encouraged to go beyond what is taught in the course in the use of this IDE.
    - **Version control:** the use of github will be encouraged for certain assignments - of course you are encouraged to go beyond the needs of this course as version control is a standard tool in software development.
- Online materials
    - **All materials (readings, videos, tutorials, quizzes, and assignments) will be online and posted on the course calendar on the respective class day at the latest; exceptions will be mentioned in class AND on the course message list.** Readings will all be fairly dense, so please search for additional resources (e.g. wikipedia, coursera lectures) as needed. All attempts will be made to provide sufficient resources for everyone.
- Course communication: this year we will be using a google group email list for group communications - set up by the instructor in the first week of class.
    - **The group name will be: datastructures-spring2019@googlegroups.com**

- Feel free to use the google group to ask questions about the material others might find useful, ask about partners for problem sets or advertize group study times, or to make comments that the rest of the class might find useful.
- Sending messages: you will be originally signed up with your luc.edu address, which you will need to send from in order to mail to the list. If you would like another addressed signed up (if you expect to send to the list often) just let the instructor know.
- The google group is primarily for timely, supplementary communication; the course calendar will be the definitive source of requirements and course expectations.

Readings, Quizzes, and Exams

Course Philosophy: In this course you will be evaluated more often than other courses, with weekly quizzes, assignments, and three exams. Consider the points given to each as a guide to the effort expected. Of particular note: quizzes will be "light-weight" and you are allowed to take them at home. Given the amount of material, it is suggested that for assignments and quizzes you focus on being succinct, and for readings you focus on the main issues.

- **Readings/Tutorial testing:** quizzes will be given based on the readings and tutorial material assigned from the previous week. You will always be given one full week to review or discuss before being quizzed. As will be clear in the first few weeks, quizzes will primarily be based on the most important aspects only.

- **Sakai quizzes:** These quizzes meant to focus students on the important aspects of the readings or lectures. **You will be allowed to take these quizzes online, as many times as you would like.** (to avoid annoying retakes for marginal benefit, if you are 80% or higher, it will be rounded up to 100% - e.g. 0.4 out of 0.5 pts) There will be no due dates for these quizzes (technically, the last day of class), but it is suggested that you finish them in the suggested period in preparation for discussions and exams.

- **Exams:** Exam days are already posted and are considered fixed. Prior arrangements in all cases can be made without loss of points, but have to be discussed. **Missed exams:** Exams cannot be missed without prior arrangements or later proof of extenuating circumstances.

Assignments

Assignments are designed to engage you in your learning, so you can begin to apply these principles in practice and tailor them to your needs.

Assignments are generally due at the end of the day one week after they are assigned, unless otherwise specified. Unlike other courses, with most assignments **you are expected to post your solutions on the Trello board that is accessible to all other students**, and you are free to observe other student solutions. However, no automated copy and paste is allowed, and direct handwritten or hand-typed copying without understanding and evaluation are obviously discouraged. As observed below in class grade points, homeworks will have minimal impact on grades - they are for learning and self-evaluation rather than grading. That being said, everyone who posts a good-faith attempt on Trello for each assignment will receive one point per assignment.

Presentations: As a survey course there are far more algorithms than we can possibly cover, many which may be particularly relevant to your interests. **Students will be expected to present one simple data structure which goes beyond what is covered in the course**. Each presentation is to be 5-10 minutes on the whiteboard or with powerpoint. These presentations will be concentrated on 1-2 days in the semester. Only a basic familiarity is expect of students observing the presentations.

Final Project: One goal of the final project is to potentially add to your portfolio for job interviews. Project proposals, progress reports, and final reviews will be part of the process. You are REQUIRED to work in groups, as this is part of the process of a full education. All people in the group, however, will have to contribute a substantial amount of coding, which generally limits sizes to 3 people unless the project goals are complex enough - ask for permission for larger groups.

Late policy: Assignments and project work can be turned in after the due date, however this places an undue burden on the instructor and any TA, especially when this policy is abused. As a compromise: there will be a reduction in points at the discretion of the instructor. Generally, the reduction will be minimal with a few weeks, and assignments may not be accepted after that - but again, it is at the discretion of the instructor and you are best served by finishing your assignments in a timely way.

Grading

Grades are determined by a simple points system, with a total of over 100 pts as the goal (1pt ~ 1%). The approximate distribution of points is given below, however, the exact scale is determined by point values given for each item. Note, due to the nature of the course, exams are the primary means of establishing your final grade, so please study appropriately prior to each exam.

- Assignments: ~10 pts (~1 pt each)
- Sakai quizzes: ~5 pts (0.5 pt each)

- Presentations: 10 pts (mean: 8.9, std dev: 0.6, min: 7.6, max: 10)
- Projects: 25 pts
    - Project presentations (5 points)
        - (mean: 4.4, std dev: 0.4, min: 3.5, max: 4.9)
    - Project final reports (15 points)
        - (mean: 14.0, std dev: 0.8, min: 12, max: 14.9)
    - Individual group evaluations (5 points)
- Exams: 60 pts
    - Exam I: 15 pts (mean: 8.7, std dev: 3.3, min: 0.6, max: 114, possible: 16.2)
    - Exam II: 20 pts (mean: 10.8, std dev: 4.8, min: 3.4, max:21, possible: 22.4)
    - Final Exam: 25 pts

Grading Scale. 90, 80, 70, 60 points with a '+/-' if within 3 percent of the border. Points needed to get each grade: A=93.0, A-=90.0, B+=87.0, B=83.0, B-=80.0, C+=77.0, C=73.0, C-=70.0, D+=67.0, D=63.0, D-=60.0. Don't expect this scale to change. If class grades are low (I expect the vast majority of students will get A's and B's), extra quizzes, homeworks, or exam bonuses will be given to add points. ***Note, the grading scale is based on points, not on percentages!*** (e.g. previous years, over 120 points were given - if you got 93 out of 120 points, you received an A!)

Extra credit: It's all about points so any credit is equivalent to "extra" credit. Assignments above and beyond the requirements may receive credit beyond 100%, but this will be rare, with an emphasis on creativity and quality, not quantity – if you think of something fun to add to an assignment – go for it! but if it requires significant extra effort, ask beforehand.

Participation Policy: Attendance is not required and there is no direct participation grading, but in the past there has been a strong correlation between engagement and accomplishment in the course - especially for those that are struggling with the material.

Academic Dishonesty: You are free to discuss anything freely and openly that is not a quiz or exam, including looking at each other's code, but you are NOT allowed to copy and paste or any thoughtless equivalent. Cheating on assignments or quizzes can result anywhere from a zero on the assignment to a zero for all assignments/quiz points for the course depending on the severity. Students caught cheating on exams will receive an F for the course and the matter will be discussed with the appropriate dean.

Students with Disabilities and Other Accommodations: Students seeking academic accommodations for a disability must meet with Services for Students with Disabilities (SSWD) to verify the disability and to establish eligibility for accommodations. Students may visit SSWD in Sullivan Center - Suite 117, call 773-508-3700, email SSWD@luc.edu or visit LUC.edu/sswd to begin the process. Students should schedule an appointment to discuss any academic

learning concerns and/or accommodations desired. Students are encouraged to contact SSWD as early in the semester as possible.