CS 240 Data Structures and Algorithms

James Madison University, Fall 2019

Your previous courses have focused on the fundamental concepts of object oriented programming that provide a solid foundation for approaching basic software development. As you progress in the field of computer science, you will tackle more complicated problems that require more sophisticated approaches. You may try to build systems for keeping track of social media contacts, building a searchable index of web pages, compressing files, detecting and blocking malware on the Internet, or finding commonalities among DNA sequences.

These larger challenges require greater focus on designing and analyzing your solution before you ever write a line of code. Otherwise, you will end up with programs that are unusably slow or crash when they run out of memory. Some problems may end up seeming impossible to implement.

In this course, you will build on your previous knowledge to develop a more thoughtful approach to software development. The techniques in this class will help you to apply a powerful problem-solving strategy, organize your data efficiently, and analyze the speed and memory constraints of your program. Along the way, you will increase your skills in coding, testing, debugging, and reading specifications.

Course & Instructor Information

Website:	`https://w3.cs.jmu.edu/kirkpams/240/f19`		
Time/Place:	M/W/F 9:05 – 9:55 AM (Section 1)		
	M/W/F 10:10 – 11:00 AM (Section 2)		
	ISAT/CS 243 (Monday and Wednesday), ISAT/CS 140 (Friday)		
Textbooks:	OpenDSA (available through Canvas)		
Instructor:	Prof. Michael S. Kirkpatrick	Email:	kirkpams@jmu.edu
Office:	ISAT/CS 223	Phone:	(540) 568-3371
Office Hours:	Mon 11 – 12:30 PM, Thu 1 - 3 PM, Fri 11 – 12:30 PM		

Course Catalog Description

Students learn to implement and analyze elementary data structures and the basic complexity classes of algorithms that use strategies such as greedy algorithms, divide-and-conquer algorithms, and backtracking algorithms. This analysis is especially applied to problems in searching, sorting, and parsing. *Prerequisites: Grade of 'C-' or better in CS/MATH 227, MATH 231, or CS 159.*

Course Structure

The concepts in this course serve as the core foundation of all areas of computer science. To help you master this material, this course integrates multiple research-supported teaching strategies that emphasize using class time to practice applying course concepts to support long-term learning. **You will need a mobile device (laptop or cell phone) in class each day. Please see me immediately if this will be a burden for you.**

Readings and OpenDSA exercises - As an introduction, each day has a specified *pre-class* reading assignment that includes embedded exercises. The reading assignments and exercises are integrated into the course Canvas site. The intention of these exercises is to practice your initial understanding of the material. Even if you are not initially successful, it is very important that you try to answer all questions before class.

Warm-ups and collaborative in-class activities - Each day will begin with a single warm-up question to review previous concepts and the rest of class time will be used for group-bassed in-class activities, including problem-solving worksheets and multiple-choice clicker questions (ConcepTests). Mini-lectures will be used to provide additional instruction and clarification as needed.

Metacognitive reflections - Each week will end with a short Canvas quiz that asks you to reflect on your learning process and how the material relates to other areas of CS and your life. These reflections are primarily for your benefit, giving you the opportunity to evaluate your understanding of the material. In addition, I use these responses to identify common points of confusion or misunderstanding.

Homeworks - Students often find the mathematical precision of certain aspects of this course to be very challenging. To help you with these concepts, there will be three written homework assignments. Each assignment will be formatted using LaTeX, which is a standard tool within the field of CS.

Labs and lab summary quizzes - A central theme of CS is to use code to develop and demonstrate mastery of theoretical concepts. In-class programming lab activities will be used to practice implementing fine details of various data structures. Each lab will be accompanied by a multiple-choice quiz on Canvas that focuses on the key ideas and understandings that working through the code should support.

Projects and analysis papers - There will be three projects that use or modify data structures to solve a real-world problem at a larger scale than the labs. These projects require analyzing the problem, making appropriate design choices, and writing test casses before attempting to implement the code. The first two projects will be accompanied by a short paper (3-5 pages) in which you will formally describe your algorithms in a pseudo-code format and analyze their time and space complexity. These papers will be formatted using LaTeX using a provided template.

Exams - There will be two in-class exams and a final exam. All exams are closed-book, closed-note.

Course & University Policies

Classroom inclusion - Mutual respect is a necessary component of the learning process. Please advise me regarding any concerns or personal circumstances (including your name's proper pronunciation, any name or gender pronouns not reflected on MyMadison, family obligations, or significant extracurricular commitments) that may be relevant to your full participation in class.

As computing professionals, we adhere to the ACM Code of Ethics and Professional Conduct (`https://www.acm.org/code-of-ethics`), which forbids discrimination and harassment of all types. If you feel someone is violating these principles (including inappropriate or demeaning jokes), it is your responsibility to take action by addressing the individual directly (if you feel comfortable doing so) or informing me. I will do my best to preserve your confidentiality while addressing the issue.

Attendance and late submissions - Unexpected life events happen to everyone. Some course components allow late submissions or dropped scores to account for this; others have less flexibility because of how they are used. If special circumstances arise that these accommodations–described in the Grading Policies and Procedures section–are insufficient, please see me as soon as possible.

Communication policy - Outside of class, Piazza (available through Canvas) and office hours will serve as the primary means of communication. All questions about course material, project instructions, due dates, etc., should be posted to Piazza, and you are strongly encouraged to respond to each other's questions instead of waiting on me to respond. Anonymous posting on Piazza will be allowed but is discouraged, as it prevents me from following up with you later, thus limiting my ability to help you learn. If anything discussed in class or lab time is not clear, you are strongly encouraged to meet me during office hours to prevent misunderstandings from piling up.

Please note that email should be used only for discussions that are not appropriate for Piazza or office hours. For instance, if you experience a personal difficulty that will impact your ability to learn in this course, please contact me via email to set up an appointment to talk. Unfortunately, I receive too much email to guarantee timely responses to questions related to course material.

Academic integrity - Students are expected to comply with the ACM Code of Ethics and Professional Conduct (`https://www.acm.org/code-of-ethics`) and the JMU Honor Code (`http://www.jmu.edu/honor/code.shtml`). The JMU Honor Code states it is a violation to "render *unauthorized* assistance to another student by knowingly permitting him or her to see or copy all or a portion of an examination or any work to be submitted for academic credit." The ACM Code prohibits "dishonest conduct" and obligates one to "credit the creators" of computing artifacts. In this course, these obligations mean:

- **Copying and/or sharing code is expressly forbidden**. This includes copying previous semesters' solutions (including your own or other students') or other external sources. It also includes posting your code publicly, such as on Github.

- It **IS** acceptable to consult other resources (*e.g.,* Stack Overflow or Geeks for Geeks) for clarifying examples but **NOT** wholesale copying of significant pieces of code. All such references **MUST** be documented explicitly within code comments.
- Extensive discussions with other students that are likely to lead to similar code must be disclosed *before and during submission* (either in person or documented in code comments); unintentional violations will be granted leniency, though a penalty may still apply.
- You are explicitly granted permission to work together on labs and homeworks. You are strongly encouraged to work together on project analysis papers. All such collaborations must be clearly identified on submissions.

Adding/dropping classes - You are responsible for registering for classes and for verifying your schedule on MyMadison. Deadlines for adding or dropping classes are available from the JMU Registrar.

Cancellations - JMU's cancellation policy (`http://www.jmu.edu/JMUpolicy/1309.shtml`) provides details regarding inclement weather and other emergencies.

Religious observance accommodations - All faculty are required to give reasonable and appropriate accommodations to students requesting them on grounds of religious observation. If you need to request accommodations, please let me know at least 2 weeks in advance.

Disability accomodations - JMU abides by Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act, which mandate reasonable accommodations be provided for students with documented disabilities. If you have a disability and may require some type of instructional and/or examination accommodations, please contact me early in the semester so that I can provide or facilitate provision of accommodations you may need. If you have not already done so, you will need to register with the Office of Disability Services, the designated office on campus to provide services for students with disabilities. The office is located in Wilson Hall, Room 107 and you may call 540-568-6705 for more information.

Grading Policies and Procedures

OpenDSA exercises - Each reading assignment has an assigned due date in Canvas, but that is solely to indicate when that material will be needed for in-class activities. You should attempt the assigned exercises prior to class, but you can complete them later if needed. All exercises completed prior to the last day of class will be given full credit.

Warm-ups and ConcepTests - Each warm-up will be worth 1 point. Each ConcepTest will be worth 2 points, with 1 point given for incorrect attempts. No make-up points are allowed, but your lowest 4 Monday or Wednesday scores will be automatically dropped. You must sign in using your e-ID to receive credit.

Metacognitive reflections - Each reflection is available beginning Friday morning and is due by Sunday at 5:00 PM. No late submissions will be accepted.

Labs and summary quizzes - Labs and summary quizzes are due on Autolab and Canvas by 5:00 PM on the day of the lab. At most three submissions are allowed per quiz and no late submissions will be accepted. Lab code can be submitted on Autolab as many times as needed. Late Autolab submissions will be accepted until 11:00 PM on the following Thursday for a 20% deduction.

Homeworks - Each homework assignment must be formatted using LaTeX. Submissions that are not formatted with LaTeX will be assigned a 20% penalty. LaTeX source files are due on Canvas and a printed copy is due in class, both by the start of class on the assigned due date.

Projects and analysis papers - For each project, you are required to submit your own test cases to Autolab one week prior to the project due date. At that point, official test cases will be made available on Canvas. All code must adhere to the coding standards described on the course project page. You are granted a total of up to three free late days for Autolab project submissions. You can use them all on a single project, or submit three projects one day late each. Beyond those three days, late submissions will be given an automatic 0. This policy will be enforced at the end of the semester, and no documentation or notice is required.

All analysis papers must be formatted using LaTeX. Submissions that are not formatted with LaTeXwill not be accepted. LaTeX source files are due on Canvas and a printed copy is due in class, both by the start of class on the assigned due date. You are encouraged to use the JMU Writing Center (`https://www.jmu.edu/uwc/`) for consultations. Analysis papers will be evaluated based on the following criteria.

	Excellent	Good	Acceptable	Needs Improvement
Requirements	Meets all requirements for a technical analysis.	Meets most specified requirements for a technical analysis.		Does not adhere to format, length, or technical audience.
Accuracy	Algorithm clearly defined; all relevant details thoroughly described.	Algorithm clearly defined; most relevant details addressed.	Algorithm not clearly defined or some important details missed.	Analysis does not adequately and clearly address technical details.
Professionalism	Excellent use of formal tone and concise language; content presentation is organized.	Mostly uses formal tone and concise language; presentation is organized.	Informal tone or excessive, vague language; content presentation is not organized.	Does not adequately demonstrate professional qualities.
Mechanics	Flawless grammar and technical language; no minor errors or typos.	Good grammar and technical language; minor errors or typos.	Reasonable grammar and technical language; needs proofreading.	Grammar or technical language need considerable revision.

Course Objectives and Weighting

The following table illustrates the mapping of course components and the relative weights of major course components.

		Major course components:					
		Readings and reflections	In-class activities	Homework	Labs	Projects	Exams
Following the successful completion of this course, students will be able to...	Write Java code that conforms to the standard Collections framework.				✓	✓	✓
	Formally quantify an algorithm's behavior through asymptotic analysis of function growth rates.		✓	✓		✓	✓
	Classify the theoretical and practical performance for algorithms on linear data.	✓	✓		✓	✓	✓
	Create a mathematical recurrence formula to characterize the behavior of a recursive algorithm.		✓	✓			✓
	Identify when advanced program implementation strategies are needed for solving problems.	✓	✓		✓	✓	✓
	Explain how tree structures improve on linear structures for a variety of data applications.	✓	✓		✓	✓	✓
	Explain the relative merits of using trees or hashing for creating lookup indexes.		✓		✓		✓
	Describe the strengths and challenges of modeling a collection of data as a graph.		✓		✓		✓
	Create or use unit and integration tests to facilitate rigorous debugging for robust software.				✓	✓	
	Write technical documentation that is professional and accurate.			✓		✓	
	Identify typical uses and real applications of specific data structures.	✓	✓		✓	✓	✓
	Analyze data structures' time and space complexity and select the best for a given problem.	✓	✓	✓	✓	✓	✓
	Accurately assess one's mastery of course material and identify opportunities for improvement.	✓	✓			✓	
	Make progress toward improving one's technical and professional skill set.	✓	✓		✓	✓	
	Weight on final course grade:	10%	10%	5%	15%	30%	30%

Tentative Schedule

The following dates are subject to change. You are responsible for being aware of announced updates.

Week	Date	Topic and Lab	Key dates
Week 1	8/26-8/30	Introduction and Collections Generics Collections	
Week 2	9/2-9/06	Algorithm Analysis and Lambdas Customizable Bags	
Week 3	9/9-9/13	Algorithm Analysis Linked Lists	**Homework 1 due** 9/11 in class
Week 4	9/16-9/20	Linear Structures Array-based Queues	**Project 1 due** 9/18 at 11:00 PM
Week 5	9/23-9/27	Recursive Algorithm Analysis Advanced Recursion Techniques	**Analysis 1 due** 9/25 at 11:00 PM
Week 6	9/30-10/4	Recursive Analysis and Sorting *[No lab this week]*	**Homework 2 due** 9/30 in class
Exam 1 – 10/4 in class			
Week 7	10/7-10/11	Sorting Customized Sorting	
Week 8	10/14-10/18	Binary Trees Expression Trees	
Week 9	10/21-10/25	Binary Search Trees Binary Search Trees	**Project 2 due** 10/23 at 11:00 PM
Week 10	10/28-11/1	Balanced Search Trees Self-balancing Trees	**Analysis 2 due** 10/30 at 11:00 PM
Week 11	11/4-11/8	Heaps and Priority Queues *[No lab - Exam 2]*	**Homework 3 due** 11/4 in class
Exam 2 – 11/8 in class			
Week 12	11/11-11/15	Hashing Hash Tables	
Week 13	11/18-11/22	Graphs Graph Algorithms	**Project 3 due** 11/20 at 11:00 PM
Thanksgiving Break – 11/25-11/29			
Week 14	12/2-12/6	Computing Ethics and Applications Applications *[2-day lab Wed and Fri]*	