

UMBC UGC New Course Request: CMSC 341J – Data Structures

Date Submitted: 2/14/2020

Proposed Effective Date: 8/23/2020

	Name	Email	Phone	Dept
Dept Chair or UPD	Jeremy Dixon	jdixon@umbc.edu	5-8866	CSEE
Other Contact	Jeannette Kartchner	jkartch@umbc.edu	5-1826	CSEE
Other Contact	Mohamed Younis	younis@umbc.edu	5-3968	CSEE

COURSE INFORMATION:

Course Number(s)	CMSC341J
Formal Title	Data Structures
Transcript Title ($\leq 30c$)	Data Structures
Recommended Course Preparation	
Prerequisite <small>Unless otherwise indicated, a prerequisite is assumed to be passed with a "D" or better.</small>	CMSC202 and CMSC 203 both with a "C" or better
# of Credits <small>Must adhere to the UMBC Credit Hour Policy</small>	6
Repeatable for additional credit?	Yes <input checked="" type="checkbox"/> No
Max. Total Credits	6 <small>This should be equal to the number of credits for courses that cannot be repeated for credit. For courses that may be repeated for credit, enter the maximum total number of credits a student can receive from this course. E.g., enter 6 credits for a 3 credit course that may be taken a second time for credit, but not for a third time. Please note that this does NOT refer to how many times a class may be retaken for a higher grade.</small>
Grading Method(s)	<input checked="" type="checkbox"/> Reg (A-F) <input checked="" type="checkbox"/> Audit <input checked="" type="checkbox"/> Pass-Fail

PROPOSED CATALOG DESCRIPTION (Approximately 75 words in length. Please use full sentences.):

An examination of a range of advanced data structures, with an emphasis on an object-oriented approach. Topics include asymptotic analysis; various binary search trees, including red-black and splay trees; skip lists as alternatives to binary search trees; data structures for multidimensional data such as K-D trees; heaps and priority queues, including binary heaps, binomial heaps, leftist heaps (and/or other mergeable heaps); B-trees for external storage; other commonly used data structures, such as hash tables and disjoint sets. Programming projects in this course will focus on implementation issues for data structures and on empirical analysis of their asymptotic performance. This version of data structures is designed for students who have a background in Java and will teach data structures in both Java and C++.

RATIONALE FOR NEW COURSE:

a) Why is there a need for this course at this time?

After a thorough analysis related to our retention rates for CMSC students, we have identified that transfer students are more than twice as likely to be unsuccessful (DFW) in our traditional CMSC 341 course than our UMBC first-time freshman population. As such, we are proposing a new version of CMSC 341 that will allow students to have additional support through this course. Most of the transfer students (especially those from neighboring community colleges) have a significant background in Java. As our CMSC 341 is taught in C++, we are hoping that a larger version would allow us to teach the material in both Java and C++. Additionally, this will provide additional support

- b) How often is the course likely to be taught?
We hope to offer it every semester at both UMBC and USG.
- c) How does this course fit into your department's curriculum?
It is a key and required course in the CMSC curriculum. Every CMSC student takes it.
- d) What primary student population will the course serve?
The course is designed for sophomore and juniors who are CMSC majors and minors.
- e) Why is the course offered at the level (ie. 100, 200, 300, or 400 level) chosen?
It is an expansion of an existing course. It has several prerequisites and it the prerequisite for almost all of our 400 level courses.
- f) Explain the appropriateness of the recommended course preparation(s) and prerequisite(s).
This course builds on concepts in CMSC 202 and CMSC 203 which has prerequisites of CMSC 201 and MATH 150/151. Therefore, students should be familiar with programming in either Java or C++ but most likely Java.
- g) Explain the reasoning behind the P/F or regular grading method.
Students can take this course for a letter grade, P/F, or audit although almost all students take it as A-F.
- h) Provide a justification for the repeatability of the course.
This course cannot be repeated.

ATTACH COURSE SYLLABUS (mandatory):

CMSC 341J: Data Structure

Prerequisites:

CMSC 202 and CMSC 203 each with a C or better.

Instructor:

Name: TBD

Office: TBD

Office Hours: TBD

Phone: TBD

Email: TBD

Course Description:

An examination of a range of advanced data structures, with an emphasis on an object-oriented approach. Topics include asymptotic analysis; various binary search trees, including red-black and splay trees; skip lists as alternatives to binary search trees; data structures for multidimensional data such as K-D trees; heaps and priority queues, including binary heaps, binomial heaps, leftist heaps (and/or other mergeable heaps); B-trees for external storage; other commonly used data structures, such as hash tables and disjoint sets. Programming projects in this course will focus on implementation issues for data structures and on empirical analysis of their asymptotic performance. This version of data structures is designed for students who have a background in Java and will teach data structures in both Java and C++.

Credits:

Six credits: not repeatable

Books:

Downey, A. (2016). Think Data Structures Algorithms and Information Retrieval in Java. Green Tea Press. (<http://greenteapress.com/thinkdast/thinkdast.pdf>). (free – required)

Goodrich, M., Tamassia, R., and Mount, D. (2011). *Data Structures and Algorithm in C++* (second edition), John Wiley & Sons, Inc.(required)

Learning Outcomes:

At the end of the course, the student will:

1. Improved problem-solving abilities, program design skills and coding skills
2. Understanding of the fundamental programming concepts of abstract data types and the object-oriented programming paradigm.
3. Analyze common operations performed on a variety of data structures using asymptotic and amortized analysis as appropriate.
4. Perform common operations on a variety of data structures using the appropriate algorithms.
5. Prove theorems regarding the performance of common operations on a variety of data structures.

6. Effectively use software development tools including IDEs and standalone debuggers
7. Use and/or modify and/or implement data structures in a complex application including recursive solutions as appropriate
8. Choose an appropriate implementation of a data structure based on application requirements.
9. Improved ability to test and debug programs

Course Format and Procedures:

This is a version of the required CMSC 341 designed for students who have a significant background in Java instead of C++. Students are required to take one of the following: CMSC 341, CMSC 341H, or CMSC 341J.

CMSC 341J is designed to teach data structures to students in two languages: C++ and Java. As the expectation is that students will have a significant background in Java, the data structures will be initially taught in Java. After the data structures are initially taught in Java, they will then be retaught in C++.

The course itself will be a combination of introducing new structures and concepts and then doing related coding examples. The course will be project heavy requiring students to code solutions in both languages throughout the semester.

Readings:

Evans, B. J., & Flanagan, D. (2019). Java in a nutshell. Sebastopol, CA: O'Reilly Media, Inc.

Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). Data Structures & Algorithms in Java. New Jersey, John Wiley & Sons, Inc.

Savitch, W. (2015). Absolute C++. 6th Edition. Pearson. ISBN: 978-0133970784

Course Topics:

Students in data structures will participate by:

- Introduction: Introduction to data structures and setting up your environment.
- Object Oriented Programming (Java and C++)
 - Classes and objects
 - Encapsulation and data hiding
 - Inheritance/Aggregation
 - Polymorphism
 - Pointers
 - Constructors, Copy Constructors, and Destructors
 - Operator overloading
 - Generics
 - Exceptions
- Data Structures Topics:
 - Asymptotic Analysis
 - Lists, stacks and queues
 - General Trees
 - k-ary trees
 - B-trees

- Binary Search Trees
- Balanced Trees
- AVL
- Red-Black
- Splay Trees
- Priority Queues using:
 - Binary heaps
 - Mergeable heaps
- Hash Tables
- Skiplists
- Union-Find for disjoint sets
- Graphs

Grading:

Assignment	Total
Projects	40%
Homework/Labs	15%
Exams	45%
Total:	100%

Grading is on a standard 10-point scale, so you will get an A for 90.0 or more total points, a B for 80.0 or more but less than 90.0 points, and so on.

The homework assignments will be a blend of practical exercises and questions that cement conceptual knowledge. The projects will be assigned both in Java and C++ and in some cases will be the same project assigned in both languages.

Academic Integrity

By enrolling in this course, each student assumes the responsibilities of an active participant in UMBC's scholarly community in which everyone's academic work and behavior are held to the highest standards of honesty. Cheating, fabrication, plagiarism, and helping others to commit these acts are all forms of academic dishonesty, and they are wrong. Academic misconduct could result in disciplinary action that may include, but is not limited to, suspension or dismissal. To read the full Student Academic Conduct Policy, consult the Academic Integrity Resources for Students page (<https://aetp.umbc.edu/ai/resources-for-students/>) or the Faculty Handbook (<http://provost.umbc.edu/faculty-handbook/>), specifically Sections 14.2-14.3.

If you need help with a project, see your instructor, your TA, or tutors provided by the Learning Resource Center. We also encourage you to consult textbooks and code examples provided on Blackboard. Consult Blackboard for additional Academic Integrity policies for projects.

Any act of dishonesty will be reported to the University's Academic Conduct Committee for further action, which may include, but is not limited to, academic suspension or dismissal from the University.

We will be using special software to check for cheating. The software is quite sophisticated and has surprised many students in the past. There is no difficulty in comparing every pair of assignments – even assignments submitted to other sections of this course, or from previous semesters.

This is a *non-exhaustive* list of restrictions for completing your assignments in this course.

- If you have questions about what is acceptable, please contact a professor or TA.

You may not look at, access, download, or obtain anyone else's work.

- You should think carefully about the assignment, and the assignment you turn in should be entirely a product of your own understanding of the material.
- You may not use any online resources to request additional help. Please contact a professor or TA for additional help.
- You may not post any part of a course document online. Posting any slides, projects, or labs will be considered a violation of this course policy and will result in an "F" for the course.
- You may not look at someone else's code "for reference," even if you put it aside before programming, and even if that person is not a CMSC student.
- You may not Google or search for the solution to an assignment, even if it's "only for reference."
- You may not copy code other than that provided in the course materials (slides, book, labs, etc.).
- You may not let someone else explain a solution to you in such detail that they are effectively dictating the code to you line by line. It does not matter if this person has never taken this course, or if they are not looking at their own code while doing so!

Student Disability Services:

UMBC is committed to eliminating discriminatory obstacles that may disadvantage students based on disability. Services for students with disabilities are provided for all students qualified under the Americans with Disabilities Act (ADA) of 1990, the ADAAA of 2009, and Section 504 of the Rehabilitation Act who request and are eligible for accommodations. The Office of Student Disability Services (SDS) is the UMBC department designated to coordinate accommodations that would allow students to have equal access and inclusion in all courses, programs, and activities at the University.

If you have a documented disability and need to request academic accommodations for access to your courses, please refer to the SDS website at sds.umbc.edu for registration information and to begin the process, or alternatively you may visit the SDS office in the Math/Psychology Building, Room 212. For questions or concerns, you may contact us through email at disAbility@umbc.edu or phone (410) 455-2459.

If you require accommodations for this class, make an appointment to meet with your instructor to discuss your SDS-approved accommodations.

Disclosures of Sexual Misconduct and Child Abuse or Neglect

As an instructor, I am considered a Responsible Employee, per UMBC's Policy on Prohibited Sexual Misconduct, Interpersonal Violence, and Other Related Misconduct (located at <http://humanrelations.umbc.edu/sexual-misconduct/umbc-resource-page-for-sexual-misconduct-and-other-related-misconduct/>). While my goal is for you to be able to share information related to your life experiences through discussion and written work, I want to be transparent that as a Responsible



Employee I am required to report disclosures of sexual assault, domestic violence, relationship violence, stalking, and/or gender-based harassment to the University's Title IX Coordinator. As an instructor, I also have a mandatory obligation to report disclosures of or suspected instances of child abuse or neglect (www.usmh.usmd.edu/regents/bylaws/SectionVI/VI150.pdf).

The purpose of these reporting requirements is for the University to inform you of options, supports and resources; you will not be forced to file a report with the police. Further, you can receive support and resources, even if you choose to not want any action taken. Please note that in certain situations, based on the nature of the disclosure, the University may need to act.

If you need to speak with someone in confidence about an incident, UMBC has the following Confidential Resources available to support you:

The Counseling Center: 410-455-2472

University Health Services: 410-455-2542

(After-hours counseling and care available by calling campus police at 410-455-5555)

Other on-campus supports and resources:

The Women's Center, 410-455-2714

Title IX Coordinator, 410-455-1606

Additional on and off campus supports and resources can be found at:

<http://humanrelations.umbc.edu/sexual-misconduct/gender-equitytitle-ix/>

Tentative Schedule:

Week	Topic	Assignment s
1	Introduction	
1	Memory & Processes	HW 1
2	Linked Lists vs Arrays	
2	Asymptotic Analysis	Project 1
3	Running Times	HW 2
3	Running Times (cont'd), STL Stacks & Queues	
4	more Stacks & Queues	
4	Trees	
5	Binary Search Trees, Project 2	Project 2
5	more BSTs	
6	AVL Trees	HW 3
6	AVL Trees, Splay Trees	
7	Midterm 1 Review	
7	Midterm Exam 1	
8	Red-Black Trees	Project 3
8	Red-Black Trees	
9	Red-Black Trees	HW 4
9	Heaps	
10	More Heaps	
10	More Heaps	Project 4
11	Hash Tables	
11	Hash Tables	HW 5
12	Exam 2 Review	
12	Exam 2	
13	Graphs	Project 5
13	Graphs	HW 6
14	<u>Case Study: The Pokegraph</u>	
14	Disjoint Sets	
15	Disjoint Sets	HW 7
15	Final Exam Review	

This schedule is subject to change without notification from the professor.