UMBC UGC New Course Request: CMSC310 – Data Analysis and Structures

Date Submitted: 3/12/2020 Proposed Effective Date: 8/23/2020

	Name	Email	Phone	Dept
Dept Chair or UPD	Jeremy Dixon	jdixon@umbc.edu	5-8866	CSEE
Other Contact	Anupam Joshi	joshi@umbc.edu	52733	CSEE
Other Contact	Mohamed Younis	younis@umbc.edu	5-3969	CSEE

COURSE INFORMATION:

Course Number(s)	CMSC310
Formal Title	Data Analysis and Structures
Transcript Title (≤30c)	Data Analysis and Structures
Recommended Course Preparation	
Prerequisite **NOTE:** Unless otherwise indicated, a prerequisite is assumed to be passed with a "D" or better.	CMSC 210 with a C or better
# of Credits Must adhere to the UMBC Credit Hour Policy	3
Repeatable for additional credit?	☐ Yes ☒ No
Max. Total Credits	3 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.
Grading Method(s)	☒ Reg (A-F) ☒ Audit ☒ Pass-Fail

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

The purpose of this course is to introduce non-computer science and non-computer engineering students to principles of algorithms and data structures. This course teaches students how to design increasingly complex programs in a manageable way, using abstract data structures, data analysis and manipulation, and other software engineering concepts. Using these data structures, programs can be designed to analyze and visualize data sets imported from external locations such as websites or databases. The classroom experience will be active combining the introduction of new topics with in-class activities.

RATIONALE FOR NEW COURSE:

a) Why is there a need for this course at this time?
This course is the third course in our applied computing sequence intended for non-computer science, non-computer engineering majors. It will be the final required course in this minor.

b) How often is the course likely to be taught?
We will most likely offer it every fall semester, increasing to every semester as the population increases.

c) How does this course fit into your department's curriculum?
It is the third and final course in our applied computing sequence intended for non-computer science, non-computer engineering majors.

d) What primary student population will the course serve?
This course will serve non-computer science, non-computer engineering majors, typically at the sophomore and junior levels.

e) Why is the course offered at the level (ie. 100, 200, 300, or 400 level) chosen?
The course builds upon and is an extension of the computing and analysis knowledge introduced in CMSC 201 – Computer Science I and reinforced in CMSC 210 – Advanced Computing. This course attempts to bridge the gap from object-oriented programming and web programming to data structures and data analysis.

f) Explain the appropriateness of the recommended course preparation(s) and prerequisite(s).
This course builds directly upon CMSC 210 – Advanced Computing. Either MATH 150 or MATH 155 provides the appropriate math preparation.

g) Explain the reasoning behind the P/F or regular grading method
Students are most likely to take this course using A-F, but on occasion a student could audit or take it P-F.

h) Provide a justification for the repeatability of the course.
This course cannot be repeated for additional credit.

ATTACH COURSE SYLLABUS (mandatory):

CMSC 310: Data Analysis and Structures

Prerequisites:

CMSC 210 with a C or better

Instructor:

Name: TBD
Office: TBD
Office Hours: TBD
Phone: TBD
Email: TBD

Course Description:

The purpose of this course is to introduce non-computer science and non-computer engineering students to principles of algorithms and data structures. This course teaches students how to design increasingly complex programs in a manageable way, using abstract data structures, data analysis and manipulation, and other software engineering concepts. Using these data structures, programs can be designed to analyze and visualize data sets imported from external locations such as websites or databases. The classroom experience will be active combining the introduction of new topics with in-class activities.

This course is organized by abstract data types which are the various kinds of data that one might encounter. There will be four projects, one for each data type. There will also be two exams and a comprehensive final. Data structures are containers to implement the abstract data types. Algorithms must produce the correct answer and run in a reasonable amount of time. We will talk about asymptotic complexity with Big O notation in which we analyze the amount of time a program takes to run as a function of the program size. Some classes will also be dedicated to discussing the projects in detail in order to facilitate a strong understanding of the way in which Algorithms and Data Structures will be applied throughout the course.

Credits:

Three credits: not repeatable

Learning Outcomes:

At the end of the course, the student will:

- Demonstrate an understanding of abstract data types that are fundamental when working with large structured datasets
- Understand the fundamentals of data structures, as well as which data structure is most applicable for a given data type.
- Evaluate performance and possible resource limitations
- Demonstrate programming skills at a level in which the student can make use of data structures to solve real-world problem
- Develop graphical/interactive programs to visualize and manipulate large datasets
- Apply data structures to manipulate datasets relevant to a variety of disciplines outside of computer

science

Grading Criteria:

Type	Points Per	Subtotal
Assignment 1-4	175	700
Midterm / Final	150	300
Total		**1000**

Grading Scale:

90% - 100%	A
80% - 89%	B
70% – 79%	C
60% - 69%	D
< 60%	F

Readings:

Levitin, A. (2012). Introduction to the design & analysis of algorithms. Boston: Pearson, ISBN: 978-0132316811

Course Topics:

- Abstract Data Types: The different kinds of data that one might encounter as they relate to data structures that are used to store and manipulate data
- Asymptotic Complexity: The study of how long an algorithm takes to run (and the spaced used) as a function of the size of the probem
- Arrays: A simple datastructure in which all of the data elements are stored contiguously in RAM
- Linked Lists: A simple datastructure which each element has a "pointer" to the next element
- Trees: A family of datastructures in which each element has a number of descendents
- Databases: Storing, accessing, and querying databases
- Data Visualization: Using visualization tools to display
- Data Analysis: Using tools (Pandas or NumPy) to analyze data
- Hash Tables: A datastructure to implement the "Associative Array" abstract data type

Academic Integrity:

Academic integrity is an important value at UMBC. 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.

More information can be found at:
https://academicconduct.umbc.edu/

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.

Tentative Schedule:

Date	Topic	Learning Style	Assigments
01/28/20	Introduction	Lecture	
01/30/20	Abstract Data Types	Lecture	
02/04/20	Project 1 Discussion	Lecture and Discussion	Project 1 Assign
02/06/20	Asymptotic Complexity #1	Lecture	
02/11/20	Asymptotic Complexity #2	Lecture And Activity	
02/13/20	Unsorted Arrays	Lecture	
02/18/20	Sorted Arrays	Lecture	
02/20/20	Analysis of Arrays	Lecture	Project 1 Due
02/25/20	Project 2 Discussion	Lecture and Discussion	Project 2 Assign
02/27/20	Linked Lists #1	Lecture	
03/03/20	Linked Lists #2	Lecture And Activity	
03/05/20	Binary Search Trees #1	Lecture	
03/10/20	Binary Search Trees #2	Lecture And Activity	
03/12/20	Binary Search Trees #3	Lecture And Activity	Project 2 Due
03/17/20	Spring Break		
03/19/20	Spring Break		
03/24/20	Midterm Review	Lecture and Discussion	
03/26/20	Midterm	Lecture	Midterm Exam
03/31/20	Hash Tables #1	Lecture	
04/02/20	Hash Tables #2	Lecture And Activity	
04/07/20	Introduction to Databases	Lecture	
04/09/20	Using SQL	Lecture and Discussion	Project 3 Assign
04/14/20	Database Optimization	Lecture	
04/16/20	Project 3 Discussion	Lecture	
04/21/20	noSQL	Lecture	
04/23/20	noSQL #2	Lecture	
04/28/20	Data Analysis #1	Lecture and Discussion	Project 3 Due
04/30/20	Data Analysis #2	Lecture And Activity	Project 4 Assign
05/05/20	Project 4 Discussion	Lecture	
05/07/20	Data Analysis #3	Activity	
05/12/20	Data Visualization #1	Lecture	
05/14/20	Data Visualization #2	Lecture And Activity	
05/19/20	Final Exam Review	Lecture and Discussion	Project 4 Due

Tentative Schedule