CMPS 201: DATA STRUCTURES
FALL 2023

Credit Hours:	3
Instructor:	Dr. Arasteh, Professor
Office Location:	Room 416, Pinchback Hall
Office Phone:	(225) 771-3727
Office Hours:	MWF 9-11, TR 8-9:30
E-mail:	davoud_arasteh@subr.edu
Textbooks:	Program Design Including Data Structures, D.S. Malik, Eighth Edition, 2018, Cengage Learning, ISBN: 978-1-337-11756-2

This course addresses Program Educational Objectives, PEO 1, PEO 2, PEO 3, and PEO 4
Program ABET Outcomes, Outcome 1, 2, and 6.

Target: 70% will perform at the level of performance in achieving ABET Outcomes 1, 2 and 6.

Program Educational Objectives: The Educational Objectives of the Computer Science Program are to produce graduates who:
PEO 1: Successfully enter the competitive job market or pursue advanced study.
PEO 2: Are proficient in identifying, formulating, and solving a wide range of computing problems.
PEO 3: Are capable of working collaboratively, and communicating effectively with team members, constituents, and the public.
PEO 4: Uphold professional and ethical responsibilities, and contribute to society through active engagement.

Program ABET Outcomes: Graduates of the program will have the ability to:
Outcome 1: Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions [PEO 1, PEO 2].
Outcome 2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline [PEO 1, PEO 3].
Outcome 3: Communicate effectively in a variety of professional contexts [PEO 1, PEO 3].
Outcome 4: Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles [PEO 1, PEO 4].
Outcome 5: Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline [PEO 1, PEO 3].
Outcome 6: Apply computer science theory and software development fundamentals to produce computing-based solutions [PEO 1, PEO 2].

Course Objectives: Students will learn to,

Course Objective 1. Create and use classes including accessors, mutators, constructors, and destructors. Define the properties of these members, such as private, protected, public, and static. Learn about abstract data types their implementation with classes. Learn about the concept of information hiding as implemented in C++.

Course Objective 2. Learn about recursive definitions, explore the base case and the general case of a recursive definition, and discover what a recursive algorithm is. Learn about recursive functions. Become familiar with direct and indirect recursion. Explore how to use recursive functions to implement recursive algorithms.

Course Objective 3. Learn about the basic properties of linked lists. Explore the insertion and deletion operations on linked lists. Discover how to build and manipulate a linked list. Implement linked lists as ADTs. Learn how to create linked list iterators and the operations on a linked list. Learn how to create unordered linked lists and ordered linked lists. Learn how to construct a doubly linked list. Become familiar with circular linked lists

Course Objective 4. Examine various stack operations. Learn how to implement a stack as an array and a stack as a linked list. Learn about infix, prefix, and postfix expressions, and how to use a stack to evaluate postfix expressions. Learn how to use a stack to remove recursion. Examine various queue operations. Learn how to implement a queue as an array. Learn how to implement a queue as a linked list. Discover how to use queues to solve simulation problems.

Course Objective 5. Learn about the various search algorithms. Explore how to implement the sequential search, binary search, comparison-based search algorithms and their performance. Learn about the various sorting algorithms. Explore how to implement the bubble sort, selection sort, insertion sort, comparison-based, quick sort, and merge sort algorithms and how they perform.

Course Objective 6. Learn about binary trees. Learn about the basic terminologies used in binary trees: left and right subtrees, path, height, level of a node, leaves, parent of a node.
Explore various binary tree traversal algorithms. Explore how to implement the basic operations on a binary tree. Learn about binary search trees. Learn how to organize data in a binary search tree. Learn how to insert and delete items in a binary search tree. Explore non-recursive binary tree traversal algorithms. Explore binary tree traversal algorithms and functions as parameters.

Course Objective 7. Learn about graphs. Become familiar with the basic terminology of graph theory. Discover how to represent graphs in computer memory. Explore graphs as ADTs. Examine and implement various graph traversal algorithms. Learn how to implement the shortest path algorithm. Examine and implement the minimal spanning tree algorithm.

Course Objective 8. Learn the three basic components of the Standard Template Library (STL): containers, iterators, and algorithms, and basic operations on vector objects. Learn how to use the copy algorithm, and range-based for loops. Explore how various containers, such as deque and list, are used to manipulate data in a program. Learn how to use iterators. Explore how to use the container adapters: stacks and queues. Become familiar with the various types of STL algorithms.

Learn about function objects: arithmetic and relational. Become familiar with insert iterators. Explore how to use various generic algorithms.

Specific goals for the course:

A. Specific outcomes of instruction:

Course Learning Outcomes [CLO]: Upon successful completion of the course, students will demonstrate the following ability:

CLO 1. Distinguish the private, protected, and public members of a class and their implementation. Examine constructors and destructors. Know about the abstract data type (ADT). Become aware of the differences between a struct and a class. Become aware of inline functions of a class and static members of a class. [ABET 1, 2, 6] [Course Objective 1]

CLO 2. Describe the function of the base case and the general case of a recursive definition, and discover what a recursive algorithm is. Analyze recursive functions with direct and indirect recursion. Analyze the use recursive functions to implement recursive algorithms. [ABET 1] [Course Objective 2].

CLO 3. Explain the basic properties of linked lists. Utilize the insertion and deletion operations on linked lists. Describe how to implement linked lists as ADTs and how to create linked list iterators. Implement the basic operations on a linked list as well as how to create unordered linked lists and how to create ordered linked lists. Construct a doubly linked list and circular linked lists. [ABET 1, 2, 6] [Course Objective 3]

CLO 4. Explain various stack operations. Implement a stack as an array, and a stack as a linked list. Learn about infix, prefix, and postfix expressions, and how to use a stack to evaluate postfix expressions. Describe queues, various queue operations and implement a queue as an array. Outline how to implement a queue as a linked list and how to use queues to solve simulation problems.
[ABET 1, 2, 6] [Course Objective 4]

CLO 5. Explain the implementation of various search algorithm, such as sequential and binary search and their performance. Describe the asymptotic notation, Big-O, used in algorithm analysis. Outline the various sorting algorithms, their implementation and performance, such as bubble, selection, comparison-based, quick and merge sort algorithms. [ABET 1, 2, 6] [Course Objective 5]

CLO 6. Outline the basic terminologies used in binary trees: left and right subtrees, path, height, level of a node, leaves, parent of a node, and various binary tree traversal algorithms. Implement the basic operations on a binary tree. Describe how to organize data in a binary search tree, and how to insert and delete items in a binary search tree. [ABET 1, 2, 6] [Course Objective 6].

CLO 7. List the basic terminology of graph theory and explain how to represent graphs in computer memory as well as graphs as ADTs. Implement various graph traversal algorithms. Implement the shortest path and the minimal spanning tree algorithms. [ABET 1, 2, 6] [Course Objective 7].

CLO 8. Recognize the three basic components of the STL: containers, iterators, and algorithms. Know the basic operations on vector objects. Explain how to use the copy algorithm, and range-based for loops. Outline how various containers, such as deque and list used to manipulate data in a program. Know various types of iterators and their use. Implement the associative containers like sets, multisets, maps and multimaps. Describe the use of container adapters like stacks queues, and the various types of STL algorithms, as well as generic algorithms. [ABET 1, 2, 6] [Course Objective 8].

Course Weekly Content:
Week   Topic
1   Introduction
2   Array Classes
3-4   Classes and Data Abstraction.

5   Recursion
6   Linked Lists

7-8   Stacks and Queues

9-10   Searching and Sorting Algorithms

11   Binary Trees

12-13   Graphs

14-15   Standard Template Library (STL)

Educational Activities:
1. Access online course materials to obtain lecture notes, instructional video, homework assignments, quizzes, project assignments, exams and related materials.
2. Study lecture notes and review solved examples.
3. Read related subject chapter.
4. Attend scheduled discussion sessions and ask questions for more clarification.
5. Complete written assignments and graded activity at the scheduled time.

Required Tools:

1. C++ Compiler.

2. Microsoft Teams, computer, webcam, microphone, internet access.

3. Moodle quiz tool

4. Moodle assignment tool

5. Watermark assessment tool

6. Word processing software for documents (Word, PDF).

Minimum Technology Requirements:
To successfully complete the course, you will need the following:

- Computer or Tablet
- Reliable internet access or data plan
- Microphone (may be built in or attached to tablet or laptop)
- Camera (may be built in or attached to tablet or laptop)
- PDF Converter

COURSE RULES AND PROCEDURES

Read the Announcements.

Make a habit of checking the Announcements forum every time you log into your dashboard. Read all announcements posted. The announcements contain helpful information on completing assignments, due dates, viewing feedback, and more. Use the tools in Moodle to keep track of your progress in this course.

Attendance policy and class participation: Students are expected to be present and on time and participate in class (whether virtual or in-person attendance) from the beginning through the end of the class period. Though attendance is strongly encouraged, merely attending classes does not guarantee a passing grade. It is every student's responsibility to make sure that he/she signs the roll during class.

Makeup test or exam: No makeup test or exam will be given except in the case of an emergency such as the student being sick, or he/she is unable to come to class due to some unforeseen event. An official proof MUST be presented to the instructor and the student is required to take the makeup test/exam as soon he/she returns to class in the following class session. Failure to comply will result in a zero (0) grade for the test/exam.

Assignment policy: Students are NOT allowed to share their assignments or to communicate during tests or exams. No late assignments will be accepted and no make-up for assignments and quizzes.

Grading Distribution:

- Students are assessed based on comprehensive quizzes, and programming assignment and exams.

- Quizzes designed based on the fundamental concepts and analytical techniques discussed in the course.
- Programing assignments are detailed analytical program design. Students will write C++ code based on the design requirement and, test the program for performance. Then interpret the result to whether or not meets the design requirements.
- Students assessed according to their solution correctness, precision, and organization.
- Midterm and final exams designed based on the fundamental concepts and analytical techniques discussed in the course.

Grade weights:

Programing Assignments (20%)
Quizzes (30%)
Midterm Exam (25%)
Final Exam (25%)

Grading Scale: A=90 - 100, B=80 - 89, C=70 – 79, D=60 – 69, F= Below 60

MOODLE ACCESS – Southern University and A&M College at Baton Rouge will use Moodle extensively in this course. Moodle is a learning management system designed to help teachers and students communicate effectively online. The course syllabus and class materials (e.g., handouts, PowerPoint slides, journal articles, assignments, readings, etc.) will be on Moodle. The student should check Moodle DAILY for all assignments submitted via Moodle. If the student has problems with his Moodle account, he/she should contact Ms. Chrisena Williams-Brown in the Division of Information Technology via email at chrisena_williams@subr.edu or via phone at (225) 771-5017.

INSTITUTIONAL POLICIES

ACADEMIC DISHONESTY – Southern University and A&M College's *Student Code of Conduct* (Code 1.1) identifies academic dishonesty as any deliberate attempt to gain an unfair advantage in academic work. Examples of academic dishonesty include cheating, falsification of information, fraud, plagiarism, and unauthorized access to academic records, providing information, material, or other assistance with knowledge that such assistance could be used in violation of the *Student Code of Conduct* or other University policies, or providing false information in connection with any inquiry regarding academic dishonesty.
Academic Dishonesty Policy - link to PDF Report Academic Dishonesty - link to PDF
ACADEMIC GRIEVANCES – Southern University and A&M College students may seek redress in academic decisions when they believe the decision is unfair or ungrounded. The academic grievance procedure provides prompt and equitable resolutions to student academic grievances. Classroom related matters should be subjected to these procedures only when the grievance cannot be settled in the ordinary course of immediate post-class discussion. The Office of the Dean of Students does not handle academic grievances.
Student Academic Grievance Procedures - link to PDF
ADA COMPLIANCE – Southern University Baton Rouge is committed to providing reasonable accommodations for students with documented disabilities in compliance with the American with

Disabilities Act (ADA). Students may request accommodations by registering with the Coordinator for Disability Services. It is important to do this early each semester because ADA accommodations are not retroactive. Upon completion of the application, the Coordinator will be responsible for forwarding the letter of accommodations to the students' professors. Accommodations will begin upon acknowledgement that the professor has received the letter of accommodations. All discussions with the faculty/student/Coordinator will remain confidential.
Contact Information
The Office of Disability Services is in Office # 246, A. C. Blanks Hall, in the center of the campus.

Southern University Office of Disability Services
Jada J. Netters, Coordinator P.O. Box 11298 Baton Rouge, LA 70813 Phone: (225) 771-3546 (V/TTDD) Email: ods@subr.edu Website: www.subr.edu/ods
The Office of Disability Services (ODS), under the auspices of the University Counseling Center, assists students in meeting their unique academic/educational, personal, vocational and social needs that would otherwise prove to be an obstacle to educational pursuits. The Office of Disability Services (ODS) provides confidential services to those students who, in this post-secondary setting, **must request and provide the necessary documentation to verify a special needs request**. Diagnostic educational evaluations/examinations are not conducted through ODS. Academic accommodations are made based on a student's documented disabilities.

Disruption of the Academic Process—The University's Student Code of Conduct (Codes 1.6 and 2.2) broadly defines disruption/obstruction as any major/minor action which obstructs, or attempts to obstruct, an official University function, such as teaching, research, administration, or other campus activity.
Disruption Policy - link to PDF Report Classroom Disruption - link to PDF

Southern University Statement of Non-Discrimination—In compliance with Title IX of the Education Amendments of 1972, Title VI and VII of the Civil Rights Act of 1964, Section 504 of the Rehabilitation Act of 1973, and other federal, state, and local laws, Southern University and A&M College forbids discriminating or harassing conduct that is based on an individual's race, color, religion, sex, ethnicity, national origin or ancestry, age, physical or mental disability, sexual orientation, gender identity, gender expression, genetic information, veteran or military status, membership in Uniformed Services, and all other categories protected by applicable state and federal laws. This commitment applies, but is not limited to decisions made with respect to hiring and promotion, the administration of educational programs and policies, scholarship and loan programs, and athletic or other College administered programs. Discriminatory acts of any kind are strictly forbidden.
Any member of the Southern University and A&M College community has the right to raise concerns or make a complaint regarding discrimination without fear of retaliation. All inquiries regarding the application of this statement and related policies may be referred to:
Cedric Upshaw, Title IX Coordinator, at (225) 771-5565.
Complaints may also be made via email at titleix@subr.edu or by visiting www.subr.edu/titleix.

Standards for Satisfactory Academic Progress—The Higher Education Act of 1965 as amended and final regulations set by the United States Department of Education (34CFR668.16) require that institutions of higher education establish reasonable standards of Satisfactory Academic

Progress (SAP) as a condition of continuing eligibility for federal aid programs. Financial aid recipients are expected to make reasonable progress as a condition of receiving and continuing to receive student financial aid. Student progress is assessed according to both qualitative and quantitative measures. The University has developed this policy to provide a framework for monitoring and determining a student's Satisfactory Academic Progress in accordance with Federal and Institutional requirements. This policy applies to all new, transfer, re-entry, re-admit with transfer work, and continuing students at Southern University.
For additional information, please visit www.subr.edu/cusa

Student Email Policy—Email is a universal service that has greatly enhanced communication both internally within the Southern University and A&M College community and externally to users, including prospective students, alumni, and the public at large. The purpose of the University's general email policy is to describe the appropriate use of University email facilities, associated responsibilities, and rights of all users of University email facilities and official Southern University and A&M College email accounts. This student email policy is a supplement to the University's general email policy.
Student Email Policy - link to PDF

Tobacco Free Campus Policy—The Southern University System (SUS) maintains a 100% Tobacco-Free Policy. Smoking and tobacco use of any kind will be prohibited on all SUS campuses and/or other properties owned and/or leased locations/premises; all internal and external areas, parking garages, and parking lots; all entrances and exits; and in all SUS owned and/or leased vehicles. Students may not smoke in their own or others' vehicles when the vehicles are parked on SUS properties. This policy will be in place at all Southern University and A&M College sponsored events-both on our premises and at external locations hosting such events, including non-University hours and will further apply to all faculty, staff, students and visitors.

SUBR Tobacco Free Policy - link to PDF