BOB JONES UNIVERSITY
LEARN. LOVE. LEAD.

CpS 210
Data Structures
Spring/2025

College of Arts and Science
Division of Mathematical Sciences

Professor:	James A. Knisely, Ph.D.
Office:	Alumni 64
Office Hours:	MWF 7:45-8:50 a.m. TTH 7:45-9:15 a.m. Other times by appointment.
Email:	jknisely@bju.edu
Telephone:	Extension 8144
Communication Policy:	For class questions that all students might benefit from, please use the class specific MS Teams team. For other types of questions or notifications, please use the chat feature of MS Teams or email. Most questions involving short answers are responded to within four hours, others within 24 hours. Please email or message if you desire a meeting so that a location can be agreed upon that allows for privacy, help.
Classrooms:	*Lecture* - AL 220; *Lab - AL 219*
Meeting:	Lecture meets MWF 2:00 - 2:50 p.m.; Lab meets Thursdays at 12:30 p.m.
Credit/Load:	3/3
Textbook and Resources:	**Print:** • *Algorithms Illustrated* series Books 1, 2, and 3; See algorithmsilluminated.org • Optional: *A Tour of C++* Updated for C++ 20 by Bjarne Stroustrup. Third Edition; ISBN 0-13-681648-7 • Optional: *The C++ Programming Language* by Bjarne Stroustrup. Fourth Edition; ISBN: 0-321-56384-0 **Online:** • Data Structures and Algorithm Analysis in C++ by Clifford Shaffer • Algorithms • Stroustrup C++ page(s), C++ 11 FAQ • SGI's STL documentation • STL Programmer's Guide

Catalog Description:

Data structures and algorithm analysis. Includes an introduction to an alternate computing platform.

Course Context:

Data Structures is a course primarily taken by computer science majors and minors. Hence, its main context is to fulfil the goals and objectives of the Computer Science Program.

Computer Science Program
It is our desire that all students in the Computer Science Major exhibit the ability to:
CS1. Design and implement solutions to practical problems
CS2. Use appropriate technology as a tool to solve problems in various domains
CS3. Create efficient solutions at the appropriate abstraction level
CS7. Demonstrate an understanding of social, professional and ethical considerations related to computing
CS8. Demonstrate understanding of fundamental concepts in the student's discipline

Course Goals:

The goals of this course are to

- improve your knowledge and experience with
  - Template and/or generic functions and classes,
  - Generic and template implementations of data structures in a variety of languages,
  - The Linux operating system, the Windows subsystem for Linux, and/or the Mac terminal

- increase your knowledge of computer science, specifically in these areas:
  - Data structures -- both fundamental and advanced;
  - Recursive algorithms, greedy algorithms, and dynamic programming;
  - Depth-First and Breadth-First Searching
- increase your awareness of social issues involving computers including privacy and civil liberties.

Course Objectives:

The student will be able to

1. Describe how the data structures in the topic list are allocated and used in memory. *Evaluated in test 1.*
2. Describe common applications for each data structure in the topic list. *Evaluated in multiple tests especially test 2.*
3. Implement the user-defined data structures in a high-level language. *Evaluated in labs 7, 10, & 11; program 4.*
4. Write programs that use each of the following data structures: arrays, records, strings, linked lists, stacks, queues, and hash tables. *Evaluated in programs 1, 2, 3 & 4.*
5. Compare and contrast the costs and benefits of dynamic and static data structure implementations. *Evaluated in test 2*
6. Choose the appropriate data structure for modeling a given problem. *Evaluated in test 3 and the final.*
7. Describe the concept of recursion and give examples of its use. *Evaluated in test 2.*
8. Identify the base case and the general case of a recursively defined problem. *Evaluated in test 2.*
9. Describe the divide-and-conquer approach. *Evaluated in test 3.*
10. Implement, test, and debug simple recursive functions and procedures. *Evaluated in labs 2, 3 & 4.*
11. Describe how recursion can be implemented using a stack. *Evaluated in lab 2.*
12. Determine when a recursive solution is appropriate for a problem. *Evaluated in test 2 and final.*
13. Explain the use of big O, omega, and theta notation to describe the amount of work done by an algorithm. *Evaluated in all chapter tests esp. test 1.*
14. Use big O, omega, and theta notation to give asymptotic upper, lower, and tight bounds on time and space complexity of algorithms.
15. Determine the time and space complexity of simple algorithms. *Evaluated in all chapter tests esp. test 1.*
16. Implement a greedy algorithm to solve an appropriate problem. *Evaluated in program 4 and lab 9.*
17. Implement a divide-and-conquer algorithm to solve an appropriate problem. *Evaluated in program 3.*
18. Solve problems using the fundamental graph algorithms, including depth-first and breadth-first search, single-source and all-pairs shortest paths, transitive closure, topological sort, and at least one minimum spanning tree algorithm. *Evaluated in programs 2, 3, & 4.*
19. Demonstrate the following capabilities: to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in programming context. *Evaluated in the Object Selection Project.* **Note: this assessment is also used in the evaluation of the CpS program.**

Course Requirements:

A 10-point grading scale, with 1-digit rounding, will be used. The grade for this class will be based upon the following categories:

Category	Points	Description
Labs	150	Ten labs worth 15 points each are scheduled.
Object Selection Project	120	Has three parts worth 20%, 40%, and 40%.
Programs	200	Four programming assignments worth 50 points each will be assigned.
Tests	300	Three tests are scheduled. Each one will cover one or more chapters.
Final Exam	80	Emphasizes the application of the knowledge gained during the semester

General Policies:

Class Deportment
Compliance with student handbook policies is expected during class. All class deportment should reflect your intention to pay attention, to be polite, and to be professional. Laptops may be used to take notes and to perform calculations and constructions during class. Please do not use the laptop for other purposes during class since studies have demonstrated that one's student's misuse of a laptop during class tends to diminish the learning of the surrounding students.

Accommodations for Students with Disabilities
Any student with disabilities or any additional needs is encouraged to contact the instructor within the first week of the course to discuss accommodations that may be necessary.

Attendance Policies and Academic Penalty for Absences

- Attendance will be tracked and reported according to the university attendance policy: BJU Policies
  - Students are expected to attend and arrive on time for all scheduled class sessions, including the final exam.
  - Students are to use effective time management in order to meet their class attendance responsibilities.

    - Arriving late or leaving early is marked as a partial attendance. Three (3) partial attendance marks count as an absence.
    - Missing a substantial of class is marked as an absence.
    - For more details and information about chronic illness, please see the Class Attendance Policy on the BJU Policies page.
- Students are responsible for all material and announcements given in class.
- If a student is absent for an exam and has a good reason, the student is to notify the instructor before the exam is covered in the next class. *Make-up exams are given at the professor's discretion. A late penalty may be applied.*

Late Work
The following policy is the standard late policy for courses taught in the Department of Computer Science:
Assignments can receive full credit only if submitted by the prescribed deadline. A 25% penalty will be applied if the assignment is not turned in on time. No credit is possible after one week past the original deadline.

The late penalty may be waived for one late assignment. Please discuss this with the professor if needed.

Academic Honesty
You are expected to uphold the school standard of conduct relating to academic honesty:

- School-wide – The link can be found on the BJU Policies page.
- CpS Department clarifications – What is allowed/disallowed in code submissions.

You must assume full responsibility for the content and integrity of the academic work you submit. The guiding principle of academic integrity is that your submitted work; examinations, reports, and projects must be your own work. You are guilty of violating this policy if you:

- Represent the work of others as your own.
- Use or obtain unauthorized assistance in any academic work.
- Give unauthorized assistance to other students.
- Modify, without instructor approval, an examination, paper, record, or report for the purpose of obtaining additional credit.
- Misrepresent the content of submitted work.

Misrepresenting your work is unethical in any setting. In an academic setting, it is a breach of the university policies. The penalty for cheating is severe. Any student cheating is subject to receive a failing grade for the assignment and will be reported to the Dean. If you are unclear about whether a particular situation may constitute cheating, consult with your instructor about the situation. For this class, it is permissible to assist classmates in general discussions of construction techniques. General advice and interaction are encouraged. Each of you must develop your own solutions to the assigned projects, assignments, and tasks. In other words, you may not "work together" on graded assignments with other students unless instructed to work as a group on a particular assignment. Such collaboration constitutes cheating. You may not use or copy (by any means) another's work (or portions of it) and represent it as your own.

Learning how to use sources appropriately is a vital part of your development as a student. To assist you in this endeavor, the university uses Turnitin, an academic plagiarism checker. Registration in this course constitutes permission for the teacher to submit any or all assignments to Turnitin.

Need Help?
You must seek help when needed because you are the only one who knows when you need it. If you need help, reach out to one of the following ways:

- Teacher – It is always best to seek help in person, either in my office or before class, if time allows. You may also text me or email me in order to set up a time in which to come see me if you have a class or are working during my announced office hours. My door is always open during my office hours. I encourage you to come see me for help.
- Classmates – Studying for tests with other students is helpful. You may work together on the labs. The four programming assignments are to be completed individually. The Academic Honesty CS-specific link above describes the help allowed for programs.

Copyright Policy:

Lecture Schedule:

- 2025 Spring Schedule
- TC: A Tour of C++
- ODS: OpenDSA Data Structures and Algorithms
- *Algorithms Illuminated* | B1: The Basics | B2: Graph Algorithms and Data Structures | B3: Greedy Algorithms and Dynamic Programming