BOB JONES UNIVERSITY
PURSUE A HIGHER CALLING

College of Arts and Science
Division of Mathematical Sciences

CpS 210
Data Structures
Spring 2026

Professor:	James A. Knisely, Ph.D.
Office:	Alumni 64
Office Hours:	MWF 7:30 – 8:45 a.m. | Th 7:30 – 9:15 a.m. Tu 1:30 – 2:45 p.m. | Please email or text to confirm availability
Email/Text/Teams:	jknisely@bju.edu | Cell 864-517-2437 | CpS 210 S26
Communication Policy:	Feel free to email or contact me via Microsoft Teams for questions and/or extended help. You may text where appropriate (not during class).
Classrooms/Meeting:	*Lecture* - MWF 2:00 in AL 302 | *Lab* - TH 9:30 in MB 203
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:

Deportment

Compliance with student handbook policies is expected during class. The classroom is to be a professional environment. That means you are to come to class prepared for the day's discussion, your attention is expected to be on course related material, and you are expected to positively contribute to the class.

Emergencies During Class

In case of emergency requiring evacuation, students will go down the stairs on the fountain side and exit the door facing Wade Hampton underneath the stairs. Students will immediately cross the street and gather by the fence with their class. If we are unable to exit the building, the professor will instruct the students on the best course of action. To be able to respond quickly to external

threats, professors may keep classroom doors locked. If you are late arriving to class, you may need to knock on the door and be let in.

Absences

BJU attendance policy is in effect (see https://home.bju.edu/bju-policies/ for details).

- Scheduled tests/quizzes should be taken before your planned absence; please contact your professor to make arrangements for doing so. You are personally responsible for getting notes from your classmates and discussing the missed material with them. You should not expect your professor to privately re-teach you the material you missed. Your professor is always available to help you with specific questions. If an unannounced quiz/assessment is taken during the class that you miss, you will NOT be allowed to make it up, and you WILL receive a zero on the assignment. Work may always be completed early (see your professor if you wish to take a test early).
- Missing an in-class test because you feel you are not prepared to take it is not acceptable. Work missed for this reason will not be made up and you will receive a zero on the assignment.
- For absences due to incapacitating illness or emergency, you should contact the instructor as soon as you realize you will not be in class to make arrangements to make up any missed work. Tests will be made up without penalty for the first occurrence. Each subsequent time a test is missed because of incapacitating illness or emergency, an additional 10% grade penalty for that test will be incurred. A 10% penalty will be assessed for a late submission of take-home tests. All late work must be made up by the next class period unless other arrangements have been made with the professor.

Presentation of Work

The goal is professional, fluent, and clear communication of what you know.

PW 1: Proper use of mathematical notation is expected. The structure of notation conveys specific meaning and should be used appropriately.

PW 2: Mathematical presentation is like grammar. There are subjects, verbs (=, ≤, >, etc.), and objects. Always write in complete sentences.

PW 3: Tests/presentations/projects are not only about what you know, but about what you can communicate about what you know so the presentation of your work/logic should always be neat, orderly, clearly defined, and with the appropriate amount of supporting detail. (Excessive steps are not required; however, answers alone are not (usually) acceptable.)

PW 4: Clearly label problems/sub-problems. Problems do not necessarily have to be worked in order but must be clearly labeled either way. Your professor will communicate their expectation on presenting problems out of order.

PW 5: Answers are to be presented as the logical conclusion of your work, not as the only important thing (e.g. at the start of the problem and/or unconnected with any justifying work).

PW 6: Follow any additional instructions given.

Your professor may refuse to accept work that does not meet the minimum presentation requirements above, or they may choose to deduct up to 10% from the assignment.

Late Policy

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. Due to the need to discourage repeated late work, a running total of lateness of assignments will be kept. No credit is possible after one week (of lateness on all assignments) past the original deadline.

Academic Integrity Policies:

The university's Academic Integrity Policy is in effect (see https://home.bju.edu/bju-policies/ for additional details). Also, for what is allowed/disallowed in code submissions, please reference the CpS Department clarifications.

Definitions of Integrity Violations

Integrity is the reflection of the character and nature of God in our actions; therefore, students will be expected to work with integrity. In academia, violations of integrity generally fall into one or more of the following categories:

- Cheating: unauthorized use or attempted use of assistance, information, or aids in any academic assignment
- Falsification: submitting work done by others, changing work after submitting an assignment, reporting false information about the completion of an assignment
- Unacceptable collaboration: working with others when not permitted, using AI to generate ideas, thoughts, or content without the explicit permission of the professor
- Facilitation of Cheating: helping another student violate academic integrity, communicating quiz/test questions to other students

- Plagiarism: the intentional or unintentional use to any degree of the ideas or words of one's source material without proper acknowledgement

All work done for this class must represent your own effort, your own understanding, and your own communication of the material.

Course Integrity Policies

If information is taken from other sources (which is at times appropriate), it always needs to be referenced and credit given where it is due. Use standard referencing techniques as taught in En 102. Solutions found on the internet are not to be copied.

- Labs: While you are encouraged to work together on the lab assignments, simply copying someone else's solution is neither useful nor acceptable. Your lab submimsion should represent your work and your understanding of the work.
- Tests (In-Class and Take-Home): No resources may be used while taking the test unless permitted by the professor. The presence of any unauthorized material on your desk, in your calculator, on your laptop, etc. while taking a test will be construed as cheating and will be dealt with as such.
  Internet/AI enabled devices or any communication devices (including but not limited to smart glasses, watches, earbuds, etc.) are not permitted to be used and should be stored out of sight during the testing period. Access these type of devices during the test will be construed as cheating and will be dealt with as such.
  Cheating on a test will likely result in a zero on the test and will be submitted to the Academic Integrity Committee.
- Programs: You are encouraged to discuss the general ideas needed to complete a program as discussed in this course with your classmates but are not permitted to work together on your program. *You may use AI resources while preparing to write your program. However, once you are ready to write your program, no AI resource should be used.*

Assignment submissions will be evaluated for plagiarism and AI usage at the discretion of the professor. If you have a question about any source you are considering using, it is wise to gain your professor's approval before using it. You are always permitted to ask your professor for help. Any help they choose to provide is acceptable.

AI Usage Policy

The goal of the assignments in this course is to learn to develop the skills covered, NOT to complete the tasks assigned. The use of AI to complete or jumpstart tasks defeats the goal of the assignments. Therefore, you may not use generative AI tools in this course for any assignment without the professor's express permission. AI tools include, but are not limited to, CoPilot, Apple Intelligence, Chat GPT, Bing Chat, Google Bard, Grok, Deepseek, Grammarly, and language translators.

Use of generative AI to develop code (such as Python or R) may be helpful during the project (each student has permission to use AI for only this purpose, other purposes require express permission). It would be wise to consult with your professor before incorporating it into your work. Reliance on AI to generate code has not yet resulted in an acceptable paper. If you do use it, you must document it as indicated above. You may NOT use AI to generate the text/discussion in your project.

Documentation of Permitted AI Use

Should an AI tool be used with permission, its use must be documented (including the tool used, a summary of the prompts provided and the portions of the assignment that were based on AI generated work). See https://style.mla.org/citing-generative-ai/ for details on citing the use of AI.

Lecture Schedule:

- 2026 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