



COSC160: Data Structures

Jeremy Bolton, PhD

Assistant Teaching Professor

Outline

- I. Welcome!
- II. Course Overview
- III. Course Expectations
- IV. Grading
- V. Assignments
- VI. Ethics
- VII. Goals

Welcome!

- COSC-160 – Data Structures
-
- **Instructor:** Jeremy Bolton, Ph.D.
- Assistant Teaching Professor
- Department of Computer Science
- **Email:** jeremy.bolton@georgetown.edu
-
- **Office Hours:** Daily hours will be entered on Canvas calendar (or by appointment)
-
- **TAs:** TBD (see Canvas calendar for office hours)

Course Summary

- This course is designed as a second year course for majors and minors and covers basic data structures and algorithm analysis. Starting with the art and science of analyzing algorithms, the main goal of this course is to learn various techniques for organizing data so that computer programs can access, modify, and delete data efficiently. Topics covered include basic data structures (for example, lists, stacks and queues), trees, hashing, heaps, disjoint sets, and graphs, self-adjusting data structures; worst-case, average-case, and amortized analysis; and basic problem solving techniques. The topics are theoretical in nature but have dramatic impact in practice.
- Credits: 3
- Prerequisites: COSC-052 and (COSC-030 or MATH-200)

Required Text

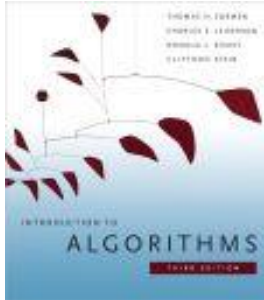


[Data Structures and Algorithms in C++ | Edition: 4](#)

Author: Adam Drozdek
ISBN: 9781133608424
Publication Date: 08/27/2012
Publisher: Cengage Learning

Optional References

- **Strongly Recommended**



[Introduction to Algorithms | Edition: 3](#)

Author: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest

ISBN: 9780262033848

Pub. Date: 07/31/2009

Publisher: MIT Press

- **Other supplemental references:**

- Data Structures and Algorithm Analysis in C++
 - Author: Mark A. Weiss
 - ISBN: 9780132847377
- Data Structures, Algorithms, and Applications in C++
 - Author: Sartaj Sahni
 - ISBN: 9780929306322
- Data Structures and Algorithms in Python
 - Author: Michael T. Goodrich
 - ISBN: 9781118290279

Course Website

- <http://jeremybolton.georgetown.domains/courses/ds/>

Notes about Coding

- **Notes about coding and coding practices:**
- Coding projects are an integral part of this course! It is assumed that you have a proficient understanding of a programming language. Students are responsible for learning and/or reviewing, as needed, the programming language chosen.
- Cheating will not be tolerated. Any form of cheating will be reported to the GU honor council. Please read the following guidelines for project submissions:
 - Discussion among students pertaining to project content and general methodology is allowed; however, students are NOT ALLOWED to share code, copy code, or use code of others without an explicit disclosure.
 - A student may be asked to present, demonstrate, or explain a project submission at any time, without notice. At my sole discretion, a student's project grade can be adjusted based on this presentation, demonstration, and/or explanation. If a student does not sufficiently understand or explain their submission, further action may be taken.
 - Due Dates will be posted in Blackboard/Canvas or announced in class.

Notes about programming Projects

- Design details will largely be left up to you.
 - Learning to ***design*** structures is an integral part of this course.
- Class discussions will largely be programming-language independent
 - I may discuss some examples and code snippets (likely in C++)

Projects

Self – Reliance : Design and Debugging

Students are largely expected to be proficient in a programming language at this point. Students are expected to review C++ resources as needed.

By the end of this course, students should have basic design skills and should be largely self-reliant for design, coding, and testing/debugging stages.

All students are encouraged to produce design documents (e.g. UML class diagrams and flow diagrams) for programming projects. *I will request to see design documents for a project before providing assistance.*

Goals of Data Structures Course

- Learn about existing Data Structures.
 - Learn about their motivations.
 - Learn how to assess them.
- Computer science is the art of problem solving. We will not only review existing data structures...
 - Cannot always simply apply existing data structures to solve ALL problems.
 - Learn to Design, Implement and Assess your own data structures for specific problems



Goals

- Learn standard data structures and concepts.
- Learn to problem solve and design structures for unique problems.
 - Learn tradeoffs in design schemes
 - Learn to design efficient solutions
- Working Independently: Programming and Debugging
- Theoretical Analysis of Structures and Associated Algorithms

Data Structures and Algorithms

- Algorithm and Data Structure concepts are interwoven
 - We will focus on the structure perspective, but algorithms will be discussed as well
- Course will have a practical and theoretical aspect