

# Advanced Data Structures and Algorithms, Fall 2020

(SENG 2000)

## Course Syllabus

Instructor:

Jonathan A. Saddler, Ph. D  
Teaching Assistant Professor of Computer Science,  
Department of Computer Science, East Carolina University  
Email: [saddlerj20@ecu.edu](mailto:saddlerj20@ecu.edu)

Class meets Mon. Wed. Fri. 2:00 PM – 3:30 PM.

Class Teams Channel: [Link](#)

Duration: Friday Oct. 2<sup>nd</sup> – Wednesday Nov. 17, 2020

### 1 Course Catalog Description

Solving computational problems using maps, hash tables, skip lists, and graph data structures. Design and analysis of algorithms using recursion, dynamic programming, and branch and bound.

### 2 Prerequisites

SENG 1020 (Data Structures for Software Engineers) is a prerequisite for this course

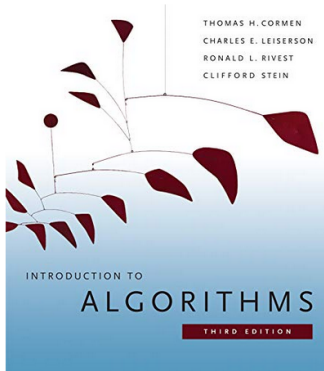
### 3 Office Hours:

I will be available for office hours at the following times, via a Microsoft teams link in our teams channel. To reach the office hours, navigate to our teams channel at [Link](#) to join the ongoing meeting.

|                  |                     |                   |
|------------------|---------------------|-------------------|
| <b>MONDAY</b>    | 10:30 –<br>11:30 AM | 3:30 –<br>4:30 PM |
| <b>TUESDAY</b>   |                     |                   |
| <b>WEDNESDAY</b> | 10:30 –<br>11:30 AM |                   |
| <b>THURSDAY</b>  |                     |                   |
| <b>FRIDAY</b>    |                     |                   |

## 4 Required Textbook

[1] Cormen, Thomas H., Leiserson, Charles E, Rivest, Ronald L., and Stein, Clifford. *Introduction to Algorithms*, Third. The MIT Press, 2009, ISBN: 978-0262033848.



### Introduction to Algorithms (Hardback)

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

★★★★☆ 6,748 ratings by Goodreads

ISBN 10: 0262033844 / ISBN 13: 9780262033848

Published by MIT Press Ltd, United States, 2009

New Condition: New Hardcover

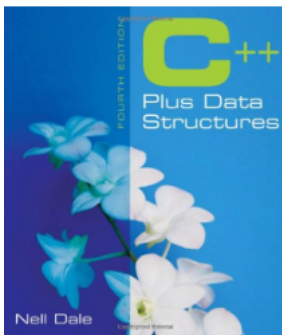
**Please use the ISBN above to link to the 3<sup>rd</sup> edition.**

It is desired that you read from the 3<sup>rd</sup> ed. for this course.

## Optional Textbook

I'll be taking examples from this book here:

[2] Dale, Nell. *C++ Plus Data Structures*. Fourth Ed. Jones & Bartlett Publishers, 2006. ISBN: 978-0763741587.



### C++ Plus Data Structures

Nell B. Dale

★★★★☆ 50 ratings by Goodreads

ISBN 10: 0763741582 / ISBN 13: 9780763741587

Published by Jones & Bartlett Publishers, 2006

New Condition: New Hardcover

Buy New

US\$ 29.99

[Convert Currency](#)

Shipping:  FREE

Within U.S.A.

[Destination, rates & speeds](#)

 Add to basket

 30 Day Return Policy

## 5 Student Learning Objectives

By the end of this course, students should know how to:

- Apply knowledge of computing and mathematics to analyze algorithms.
- Apply knowledge of computing and mathematics to design algorithms using recursion, dynamic programming, and branch and bound paradigms.
- Solve computational problems using lists, two dimensional arrays, maps, hash tables, skip lists, and graph data structures.
- Apply important software development paradigms to deploying codebases that rely on shared code across classes.

## 6 Course Topic Outline

- Sorting Algorithms and Searching Algorithms using Arrays
- Introduction to Programming Data Structures in C++
  - Variables, expressions and statements
  - Control Flow, Functions, Recursion, and Arrays
  - Freeing Up Memory
- Algorithmic Runtime Analysis and the Master Method
- Special Data Structures in C++, including...
  - Lookup Tables (Dictionaries and Maps) and Skip Lists
  - Graphs and Trees
- Graph and Tree Algorithms
- “Memo”ization in Dynamic Programming/Branch-and-Bound Algorithms
- If Time:
  - Greedy Algorithms
  - Divide and Conquer Algorithms

## 7 Course Assignments

Two types of assignments are to be submitted by students in this course.

**“Pseudocode and Problem-solving” assignments** will require that you understand the algorithm topic of the week well enough to devise solutions that utilize them to devise improvements to these algorithms. The assignments require an effort to take what was learned about algorithms in class to a further next step.

**“Complete-a-program” assignments** will ask you to utilize data structures that you have learned in class to implement either a full requirements specification or the remaining half of a partial implementation of one. The language that shall be used to turn in assignments for this course will be **C++**, a programming language that supports abstractions and object-oriented paradigms. As part of the course lecture material, we will give students an opportunity to learn how to design classes and perform operations on data using C++. “Complete-a-program” assignments go a step beyond “Problem-solving” assignments by writing computer programs that mimic the pseudocode using C++.

**Exams** are a third component of this course that will involve the recollection of specific definitions of terms defined among what you learned about algorithms. You will demonstrate your knowledge of the structure of an algorithm, as well as the abstractions present in an algorithm, what data structures are well-known to support each algorithm, and report on strengths and weaknesses of these algorithms. There are three exams to be given in this course.

## Breakdown:

\* Note: The heaviest weighted item in this course are the “complete-a-program” assignments.

| Component   | Relative Weight<br>in Final Grade           |
|---|---|
| Participation   | 8%  |
| Pseudocode and Problem-Solving Assignments                              | 20%   |
| at least 3 Pseudocode and<br>at most 2 Problem solving assignments      |   |
| Complete-a-Program Assignments*   | 32%   |
| At least 4 complete-a-program<br>assignments with increasing difficulty | potential breakdown:<br>2%, 6%,<br>12%, 12% |
| Exam 1  | 12.5%                                       |
| Exam 2  | 12.5%                                       |
| Final Exam (Wednesday, Nov. 17th, 2020)                                 | 15%   |

## 8 Grading Scale

Cutoffs for grades are listed below. Each value to the right is the lowest score that may be used to earn the letter grade on the left.

|           |     |
|-----------|-----|
| <b>A</b>  | 93% |
| <b>A-</b> | 90% |
| <b>B+</b> | 87% |
| <b>B</b>  | 83% |
| <b>C+</b> | 76% |
| <b>C</b>  | 72% |
| <b>C-</b> | 68% |
| <b>D+</b> | 64% |
| <b>D</b>  | 60% |
| <b>D-</b> | 56% |

## 9 Participation

(to be covered in course lecture)

Participation is very important in this course. Each week from here until the 17<sup>th</sup> will have a certain number of days attached to it. All participants in the course up until the add-drop date start out in tier 8. Missing a certain amount of class after that point will lower your participation grade. Exams also count as a day earned toward a 100% participation grade.

Missing enough participation points will keep you from earning that well-famed A, and will keep me from fairly being able to assign groups for group class projects (you may get assigned to a group of 1 if you miss class).

Please inform me of all intended absences at least 5 hours in advance of the lecture.

## 10 Academic Integrity Policy for SENG 2000:

“Copied code” is a bad thing and is a more self-explanatory term for “plagiarism.” When I see copied code, I will grade the assignment as usual, however, you will receive a notification for plagiarism. According to ECU Faculty Manual Part 6 on Academic Integrity Violations (below), I must pursue a specific series of actions such as special follow-up meetings to protect the integrity of all assignments submitted to our courses.

This is not a fun thing to do for either one of us. The meeting according to guidelines must happen on a strict time schedule which could interrupt your regular course schedule, and the punishment we levy could be a 0% on the assignment, a 0% or multiple assignments, or in harsher cases a failing grade for the course. Citing code that you find posted elsewhere or in someone else’s assignment will not suffice as an excuse to copy code, and such code will still be considered plagiarism if submitted.

Academic honesty is critical for success in this course. Here is a section of the faculty manual that discusses how we treat and locate plagiarism:

ECU Faculty Manual, section on Academic Integrity:

<http://www.ecu.edu/cs-acad/fsonline/customcf/currentfacultymanual/part6section2.pdf>

Plagiarism basically is submitting the work of others as your own. You may work together on homework (and are encouraged to), but you must submit your own work.

I must add an additional statement from the department statement on academic integrity below:

“Violations may include copying homework solutions from the posted solutions with no evidence of independent thought and submitting the work of other students as your own.”

*In this course we must make an exception to this.* If there are “starter solutions” that you find are posted on our course website, (Canvas), you may freely copy this code to your own solution and publish it with your solution. In these situations, it is especially important to be careful about doing your own work, and not copying still others’ solutions, as files that contain starter code may still contain “copied code” inside, around, and calling starter code procedures.

## **11 Disabilities:**

East Carolina University seeks to comply fully with the Americans with Disabilities Act (ADA). Students requesting accommodations based on a disability must be registered with the Department for Disability Support Services located in Slay 138. The contact number for Disability Services is 252.737.1016 (Voice/TTY).

## **12 Inclement Weather**

In the event of a weather emergency, information about the status of ECU classes and offerings can be obtained through the following online and dial-in sources.

ECU Emergency Notices <http://www.ecu.edu/alert>

ECU Emergency Information Hotline (252) 328-0062