

## About this course

The focus of this course is how data structures and algorithms impact the computational complexity (time and space) of programs. It is assumed that you have some familiarity with the data structures of stacks, queues, linked lists, and binary search trees (BSTs). This course additionally covers the data structures of heaps, hash tables, balanced variants of BSTs (for example, red-black, 2-3, or splay trees), and graphs (undirected and directed, weighted and unweighted). Algorithmic techniques covered include divide-and-conquer, dynamic programming, and greedy. While most algorithms are analyzed in the worst or average case, also of interest are lower bounds, and amortized and probabilistic analyses.

*Specific topics covered include:*

- Introduction to Essential Data Structures
- Implementation of Data Structures
- Introduction to Algorithms and Algorithm Strategies
- Introduction to Run Time Analysis and Big-O
- Analysis of Algorithms
- Implementation of Algorithms

## Required prior knowledge and skills

None

## Learning Outcomes

*Learners completing this course will be able to:*

- Define data structures such as heaps, balanced trees, and hash tables.
- Explain how to use a specific data structure in modeling a given problem
- Identify, construct, and clearly define a data structure that is useful for modeling a given problem.
- State some fundamental algorithms such as merge sort, topological sort, Prim's and Kruskal's algorithm, and algorithmic techniques such as dynamic programming and greedy algorithms.
- Use a specific algorithmic technique in solving a given problem.
- Design an algorithm to solve a given problem.
- Define the notions of worst-, best-, and average-case running times of algorithms.
- Analyze and compare different asymptotic running times of algorithms.
- Analyze a given algorithm and determine its asymptotic running time.
- Combine fundamental data structures and algorithmic techniques in building a complete algorithmic solution to a given problem.

## Estimated Workload/Time Commitment Per Week

- About 5 to 7 hours per module
- 30 to 60 minutes per practice quiz
- 2 hours for Practice Final Exam
- 2 hours for Final Exam

## Textbook

Introduction to Algorithms, Third Edition, By Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. c 2009 by Massachusetts Institute of Technology. ISBN 978-0-262-03384-8. This textbook is your primary resource for this course. It is an industry standard and authoritative text on the subject. It is strongly recommended that you should read the first six sections (first 26 chapters) of this textbook page for page. While the text will work as a reference book, its value as such is severely limited if you are not fluent in the material covered in the previous sections and chapters. Your success on the Final Exam will be directly related to your fluency with the topics covered in this text. In addition to the required text, there are many very good resources available online. Wikipedia is one excellent recourse for technical information regarding specific data structures and algorithms and data structures and algorithms in general. Participants in this course are highly encouraged to seek out and use additional resource - both books, and online materials - as they build their mastery of the topics.

## Course Completion and Outline

You will need to pass the final exam with an 80% or better, and successfully pass proctoring requirements for the final exam.

### Module 1: Introduction to Essential Data Structures

- 1.0: Overview
- 1.1: Getting Started
- 1.2: Quick Tour of Fundamental Data Structures
- 1.3: Visualizing Data Structures
- 1.4: Summary
- 1.5: Practice Quiz - Essential Data Structures

### Module 2: Implementation of Data Structures

- 2.0: Overview.
- 2.1: Fundamental Structures and Implementation
- 2.2: Implementation of Specific Data Structures
- 2.3: Summary
- 2.4: Practice Quiz - Implementation of Data Structures

### Module 3: Introduction to Algorithms and Algorithm Strategies

- 3.0: Overview
- 3.1: Algorithm Strategies
- 3.2: Overview of Specific Algorithms
- 3.3: Summary
- 3.4: Practice Quiz - Introduction to Algorithms and Algorithm Strategies

### Module 4: Introduction to Run Time Analysis and Big-O

- 4.0: Overview
- 4.1: Asymptotic Run Time Complexity
- 4.2: Big-O Notation
- 4.3: Summary
- 4.4: Practice Quiz - Introduction to Run Time Analysis and Big-O

### Module 5: Analysis of Algorithms

- 5.0: Overview
- 5.1: Algorithm Analysis
- 5.2: Summary
- 5.3: Practice Quiz - Analysis of Algorithms

## Module 6: Implementation of Algorithms

- 6.0: Overview
- 6.1: Implementation of Specific Algorithms
- 6.2: Summary
- 6.3: Practice Quiz - Implementation of Algorithms

## Module 7: Practice Final Exam

- Practice Final Exam

## Course Completion

- Final Exam Proctoring Setup
- Final Exam - Proctored

## Creators



### **Ryan Meuth, Lecturer**

In 2013, Ryan Meuth joined Arizona State University in the School of Computing, Informatics, and Decision System Engineering as a lecturer, focusing on Freshman engineering and the first year experience. Since then, he has additionally become the course coordinator for the CSE Capstone Program, ASU101-CSE, and FSE100-CSE. Meuth teaches FSE100-CSE, CSE Capstone, and a variety of CSE program courses focusing on embedded systems. Meuth was awarded the Top 5% Teaching Award in 2014, 2015 and 2016, as well as the Spirit of EPICS award in Spring 2015. Meuth's research interests are focused on the computer science classroom experience, and how that can be improved for all students.



**Phillip Miller, Lecturer**

Phill Miller joined the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University in 2014 after teaching computer science and software engineering courses for 14 years at the University of Advancing Technology in Tempe, AZ.