CSE 332 Data Abstractions

Credits
4.0 (3 hrs lecture, 1 hr section)

Lead Instructor
Ruth Anderson

Textbook

- *Data Structures & Algorithm Analysis in Java*, Weiss

Course Description
Covers abstract data types and structures including dictionaries, balanced trees, hash tables, priority queues, and graphs; sorting; asymptotic analysis; fundamental graph algorithms including graph search, shortest path, and minimum spanning trees; concurrency and synchronization; and parallelism. Not available for credit for students who have completed CSE 373.

Prerequisites
either CSE 311 or CSE 321.

CE Major Status
Required

Course Objectives

1. *communicate* the representation and organization of data in terms of ubiquitous computing abstractions such as stacks, queues, trees, hash-tables, and graphs
2. *analyze* algorithms for correctness and efficiency, including the use of asymptotic analysis
3. *design* parallel programs that use extra computational resources to complete a task more quickly
4. *recognize* software errors related to concurrent execution of tasks such as race conditions
5. *create* software that implements classic data structures and algorithms and uses such algorithms appropriately

ABET Outcomes
(a) an ability to apply knowledge of mathematics, science, and engineering
(b) an ability to design and conduct experiments, as well as to analyze and interpret data
(c) an ability to design a system, component, or process to meet desired needs within realistic

constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
(e) an ability to identify, formulate, and solve engineering problems
(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Review of simple abstract data-types (2 lecture hours)
- Algorithm Analysis (1 lecture hour)
- Asymptotic Analysis (2 lecture hours)
- Priority Queues (2 lecture hours)
- Dictionaries (5 lecture hours)
- Sorting (3 lecture hours)
- Graphs (4 lecture hours)
- Parallel Programming (4 lecture hours)
- Concurrent Programming (3 lecture hours)