CSE 101
Algorithms and Abstract Data Types
Fall 2019

Description: Studies basic algorithms and their relationships to common abstract data types. Covers the notions of abstract data types and the distinction between an abstract data type and an implementation of that data type. The complexity analysis of common algorithms using asymptotic (big "O") notation is emphasized. Topics include sorting and searching techniques, basic graph algorithms, and algorithm design techniques. Abstract data types covered include priority queues, dictionaries, disjoint sets, heaps, balanced trees, and hashing. Familiarity with C, Java, and Unix is assumed. (Formerly CMPS 101.)

Time and Place: MWF 4:00pm-5:05pm Classroom Unit 2
Class Webpage: https://classes.soe.ucsc.edu/cse101/Fall19/

Instructor: Patrick Tantalo http://users.soe.ucsc.edu/~ptantalo/
Office: E2 239A
Office Hours: MWF 12:00-2:00pm
Email: ptantalo@soe.ucsc.edu

Teaching Assistants:
Saeed Kargar (skargar@ucsc.edu)
Tuan Tran (atran18@ucsc.edu)
Trung Nguyen (tnguy404@ucsc.edu)
Rongwen Zhao (rzhaol7@ucsc.edu)
Anthony Koutroulis (aakoutro@ucsc.edu)
Spencer Neuschmid (sneuschm@ucsc.edu)

MSI Learning Assistants:
Yasin Aikebar (yaikebai@ucsc.edu)
Nixon Duong (niduong@ucsc.edu)

LSS Small Group Tutor:
Gigi Bachtel (gbachtel@ucsc.edu)

Required Text:

<table>
<thead>
<tr>
<th>Week</th>
<th>Sections</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1-1.2, handouts</td>
<td>ADTs, Analysis of Algorithms</td>
</tr>
<tr>
<td>2</td>
<td>2.1-2.3, 3.1-3.2, handouts</td>
<td>Asymptotic Growth Rates</td>
</tr>
<tr>
<td>3</td>
<td>4.3-4.5, handouts</td>
<td>Induction Proofs, Recurrences</td>
</tr>
<tr>
<td>4</td>
<td>B4, B.5 handouts</td>
<td>Graphs, Trees</td>
</tr>
<tr>
<td>5</td>
<td>22.1-22.5</td>
<td>Graph Representations, BFS, DFS</td>
</tr>
<tr>
<td>6</td>
<td>6.1-6.5</td>
<td>Heaps, Heapsort, Priority Queues</td>
</tr>
<tr>
<td>7</td>
<td>21.1-21.3, 23.1-23.2</td>
<td>Disjoint Sets, Minimum Weight Spanning Trees</td>
</tr>
<tr>
<td>8</td>
<td>24.1, 24.3</td>
<td>SSSP Problem, Bellman-Ford and Dijkstra’s Algorithms</td>
</tr>
<tr>
<td>10</td>
<td>7.1-7.4, 8.1-8.4</td>
<td>Sorting Algorithms</td>
</tr>
</tbody>
</table>
Coursework:
- 5% Homework: Written exercises, mostly from CLRS, submitted through CrowdGrader
- 35% Programming Assignments: Five projects due at roughly 10 day intervals
- 15% Midterm Exam 1: Wednesday October 23, 4:00-5:05pm
- 15% Midterm Exam 2: Wednesday November 20, 4:00-5:05pm
- 30% Final Exam: Wednesday December 11, 4:30–6:30pm

In addition, it is required that students earn a passing grade in both the Theory (Homework, Exams), and Programming portions of the course in order to receive a grade of C or better. Specifically, if either the combined Homework-Midterm1-Midterm2-Final average (weighted as above), or the Programming Assignment average, is not at least passing (≥ 70%), then the student's maximum possible grade in the course will be C-. All scores (Homework, Program, Midterm 1, Midterm 2, Final and Overall) will be rounded to the nearest 10th of a percent. They will not be rounded further. No scores in this class are curved.

Grading scale:
- A+ 97.0% - 100%
- A  93.0% - 96.9%
- A-  90.0% - 92.9%
- B+  87.0% - 89.9%
- B  83.0% - 86.9%
- B-  80.0% - 82.9%
- C+  76.0% - 79.9%
- C  70.0% - 75.9%
- C-  67.0% - 69.9%
- D+  64.0% - 66.9%
- D  61.0% - 63.9%
- D-  58.0% - 60.9%
- F  0% - 57.9%

Accommodations for Students with Disabilities
UC Santa Cruz is committed to creating an academic environment that supports its diverse student body. If you are a student with a disability who requires accommodations to achieve equal access in this course, please submit your Accommodation Authorization Letter from the Disability Resource Center (DRC) to me privately during my office hours (or by appointment), preferably within the first two weeks of the quarter. At that meeting, we will discuss how to best ensure your full participation in the course. I encourage all students who may benefit from DRC services, or who wish to just learn more about those services, to contact DRC by phone at 831-459-2089, or by email at drc@ucsc.edu. See also https://drc.ucsc.edu/.

Academic Honesty:
The Baskin School of Engineering has a zero tolerance policy for any incident of academic misconduct. If cheating occurs, consequences may range from getting zero on a particular assignment to failing the course. In addition every case of academic misconduct is referred to the students’ college Provost, who sets in motion an official disciplinary process. Cheating in any part of the course may lead to failing the course, suspension or dismissal from the Baskin School of Engineering, or from UCSC.

What is cheating? In short, it is presenting someone else’s work as your own. Examples include copying another students’ homework, programming assignment, or exam solution; allowing your own work to be copied; or in any way facilitating misconduct by others. You may discuss programming projects with fellow students, but your collaboration must be at the level of ideas only. You may freely give and receive help on the UCSC computer facilities, code editors and IDEs, the UNIX operating system, and on the proper use and syntax of the C programming language. You may also freely use any example code that is posted by
me on the class webpage. However, you may not *copy, paste, email, transfer or share* in any way the *source code* for projects in this class.

Go to [https://www.ue.ucsc.edu/academic_misconduct](https://www.ue.ucsc.edu/academic_misconduct) to see the University's official policy on Academic Misconduct.