Data Structures
CPSC 131-02/06
Spring 2018

Instructor

Dr. Anand Panangadan
Phone: 657-278-3998
Email: apanangadan@fullerton.edu (Email is the best way of reaching me)
Office: CS 538
Office Hours:
   Tuesdays 4:00 - 5:30 pm
   Thursdays 4:00 - 5:30 pm
   & by appointment. During final exam week, office hours are by appointment only.

Meeting Information

131-02: TuTh 5:30 - 6:45PM, E 202
131-06: TuTh 2:30 - 3:45PM, CS 110A

Final Exam Dates

131-02: Thursday, May 17, 5:00 - 6:50pm, same room as lecture (E 202)
131-06: Thursday, May 17, 2:30pm - 4:20pm, same room as lecture (CS 110A)

Surveys: In order to facilitate continuous improvement and course redesign, there may be periodic surveys posted to the class by both the Office of the Chancellor and the instructors.

Description & Objectives

Classical data structures: vector, linked list, stack, queue, binary search tree, and graph representations.
Prerequisites
CPSC 121 or sufficient score on the Computer Science Placement Exam.

Important Dates
CSUF’s Academic Calendar is posted online at «http://apps.fullerton.edu/AcademicCalendar/». The Academic Calendar contains all the campus closures and holidays you should be aware of. CSUF’s Admissions Calendar is posted online at «http://www.fullerton.edu/admissions/Resources/Calendars.asp». The Admissions Calendar contains all the major dates with respect to adding, dropping, and withdrawing from your classes. The final exam schedule is determined by Admissions and Records. Makeup exams are only available by advance request for documented exceptional circumstances.

Textbooks

Required

Suggested
1. Bloom’s taxonomy diagram (web)
2. Starting Out with C++: Early Objects (8th Edition), Gaddis et al.
3. eplusplus.com
4. Open Data Structures (in C++), Morin

Many popular technical books may be read online through the campus’s subscription to Safari Books Online. From outside of the campus network, the campus library’s WWW proxy will grant you access, «http://www.library.fullerton.edu/asp/ipcheck.aspx?url=http://proquest.safaribooksonline.com/?uicode=calstate». The Safari Books Online service can be accessed directly from any computer on the campus network, «http://proquest.safaribooksonline.com/».

Learning Goals
1. Analyze an algorithm or procedure and derive its time efficiency class in terms of asymptotic notation.
2. Design and/or implement software that makes effective and appropriate use of fundamental data structures (e.g. stack, queue, search tree, hash table).
3. Identify possible solutions to a problem and analyze their feasibility or trade-offs.
4. Write syntactically-correct source code, making appropriate use of fundamental constructs such as variables, branches, loops, and functions that solves a well-posed computational problem.

G.E. Requirements
This class does not meet any CSU General Education requirements.

Course Outline (Subject to change)

<table>
<thead>
<tr>
<th>Week</th>
<th>Material</th>
<th>Textbook Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>introduction&lt;br&gt;evaluating and analyzing code viz. Bloom’s taxonomy&lt;br&gt;review pointers, arrays, references, dynamic memory</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>review OOP principles&lt;br&gt;C++ classes, templates, exceptions</td>
<td>2.1, 1.5, 2.3, 2.4</td>
</tr>
<tr>
<td>3</td>
<td>arrays&lt;br&gt;singly linked lists (SLL)</td>
<td>3.1, 3.2</td>
</tr>
<tr>
<td>4</td>
<td>doubly linked lists (DLL)&lt;br&gt;DLL iterators</td>
<td>3.3, 6.2.3, 6.2.5</td>
</tr>
<tr>
<td>5</td>
<td>review recursion, recursive list operations&lt;br&gt;experimental analysis, asymptotic analysis</td>
<td>3.5, 4.2.1; 4.2.3-4.2.5</td>
</tr>
<tr>
<td>6</td>
<td>wrapper classes; stacks: linked list, fixed array, STL&lt;br&gt;queues: doubly linked list, circular array, STL</td>
<td>5.1, 5.2</td>
</tr>
<tr>
<td>7</td>
<td>dynamic vector, dynamic stack; dynamic queue&lt;br&gt;amortized efficiency, dynamic vector analysis</td>
<td>6.1</td>
</tr>
<tr>
<td>8</td>
<td>Trees, tree traversal algorithms&lt;br&gt;Binary trees, binary search trees; comparable objects</td>
<td>7.1, 7.2, 7.3.1-7.3.4</td>
</tr>
<tr>
<td>9</td>
<td>midterm exam (covers material from weeks 1-7)</td>
<td></td>
</tr>
<tr>
<td>Spring Recess</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 10 | Maps  
  Binary search trees, recursive BST search and insertion  
  recursive BST removal; | 9.1-9.1.3  
  10.1 |
| 11 | balanced BSTs; AVL tree introduction, AVL tree insert, remove, and restructuring | 10.2 |
| 12 | graphs; edge list, adjacency matrix  
  adjacency list | 13.1-13.2.1, 13.2.3  
  13.2.2 |
| 13 | depth-first search  
  breadth-first search | 13.3.1-13.3.3  
  13.3.5 |
| 14 | Hash tables | 9.2 |
| 15 | catch-up/review |
| finals | final exam (covers material from weeks 8, 10-15) |

### Technical Proficiency

Students are expected to be intimately familiar with their development platform of choice and be able to write and debug code in C++ at a level of proficiency that corresponds to the prerequisites of the course.

Technical proficiency with information technology, such as, but not limited to, the use of web-based online services, sending and receiving electronic mail, and desktop computer file systems, is assumed.

### Grading

Except for Prof. Bein’s section (131.03), plus and minus grading is not used when determining final grades. Prof. Bein will be using plus and minus grading for determining the final grades. Please see the addendum to the syllabus for section 131.03.

Final grades are computed by first finding the average score in each category described in the table below. All scores are normalized to a scale of 0 to 100 before being averaged. The average score for each category is then used to compute the weighted average according to the weights in the second table below.

<table>
<thead>
<tr>
<th>Grade</th>
<th>% of Total Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90–100%</td>
</tr>
<tr>
<td>B</td>
<td>80–89%</td>
</tr>
</tbody>
</table>
C  70–79%
D  60–69%
F  Below 59%

<table>
<thead>
<tr>
<th>Category</th>
<th>% of Final Grade</th>
<th>Drop Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>12%</td>
<td>Drop 2 lowest</td>
</tr>
<tr>
<td>Class participation</td>
<td>3%</td>
<td>N/A</td>
</tr>
<tr>
<td>Projects</td>
<td>30%</td>
<td>N/A</td>
</tr>
<tr>
<td>Reading assignments</td>
<td>5%</td>
<td>Drop 2 lowest</td>
</tr>
<tr>
<td>Midterm</td>
<td>25%</td>
<td>N/A</td>
</tr>
<tr>
<td>Final</td>
<td>25%</td>
<td>N/A</td>
</tr>
</tbody>
</table>

I reserve the right to decrease these boundaries, but will not increase them. In other words, I have leeway to adjust the boundaries downward, awarding higher letter grades, to compensate for assignments that were too difficult. In general I do not adjust grade boundaries unless the unadjusted class average is significantly lower than the department's GPA average.

**Grade calculations**

I record assignment grades in Titanium. Please check them for accuracy weekly. Titanium calculates your grade automatically. These calculations are based only on the grades that are currently available. So, for example, the grade calculation will ignore the project category until I've graded the first project. Titanium automatically drops low scores where appropriate, but only once we're far enough along that at least one score will be counted. Every project score will is weighted equally, regardless of the number of points available on each project.

**Grade exceptions**

Too often I am approached at the end of the term by students telling me how desperate they are to get a passing grade because they’re graduating or on academic probation. In these cases, there's nothing that can be done. If this class is important to you and there is a lot riding on your grade, it is your burden to work hard, come get help when necessary, attend class, complete all the projects, and do well on exams. Your obligation begins on day one. Please note that there is one syllabus for the course; all students are graded based on the requirements outlined in the syllabus, and nothing more. There are no special deals, relaxed standards or extra
opportunities based on class standing or other factors. Your grade is a function of your graded work, and that alone. That's an essential part of a fair grading system.

If you are surprised by your grade at the end of the semester, you have the right to ask if the grade was given in error. I am happy to check your scores to verify that no clerical error was made; these errors are extremely rare, but possible. In the exceptional circumstance of a clerical error, it will be corrected promptly. Note that final course grades are non-negotiable, and University policy establishes that grades are given at the sole discretion of the faculty member. If your grade was not given in error, that is your final, non-negotiable grade.

Assignments

Programming and written assignments will be discussed in class and posted to the course website in advance of their due dates.

All programming assignments must be written in the C++ programming language, unless specified otherwise. Coding style must conform to professional norms. At a minimum, code must be commented, have descriptive names for identifiers, and contain a comment with pertinent information such as the student's name, email address, and assignment name. A plain text README.TXT must be included with each assignment submission summarizing and documenting the work submitted.

At the start of the semester, the instructor will detail the platform and tools used to grade student assignments. It is the student's responsibility to ensure that the assignments execute to his or her satisfaction on the instructor's grading platform.

Homework

Homework will be due every week, except for the first, last, and exam weeks. Homework can be done in groups. Each assignment will be three problems. You should do all the problems but only one will be graded for credit (you will not be told which one will be graded). Homework must be submitted as hardcopy in class.

Textbook reading assignments

Short online tests on the reading assignments described in the Course Outline will be given on Titanium every week. The test will be due before class.

Projects

There will be four projects spaced at least two weeks apart. These are group projects. Each project will involve designing, implementing, and analyzing a substantial C++ program.

Obtaining and submitting project code

Student project code will be evaluated and graded for correctness using an automated process based on a GNU/Linux environment. Project code will be collected using GitHub «www.github.com», a web-based code management service.
We will be using GitHub Classroom to distribute starter code and collect your submissions. This requires you to have an account on github.com. If you are new to GitHub, do the following to get started:

1. Create an account at github.com. Students should create a free account on github.com with their csu.fullerton.edu email address (if you already have a GitHub account, you can choose to use that provided your csu.fullerton.edu email address is registered with it). You may want to use this account to show a portfolio of your work to prospective employers in the future, so choose something professional.
2. Read Understanding the GitHub Flow and Hello World at GitHub Guides.
3. Read the instructions below for your chosen development environment (Visual Studio on Windows/Linux)

Once you understand the basic operation of git, click the given assignment link to create your own copy of the skeleton code to your PC. One student from a group forms a new team. Here is a video that shows the steps: https://www.youtube.com/watch?v=-52quDR2QSc

Development environments

The test platform is Linux with the clang++ -std=c++11 compiler. For this reason, the recommended development platform is Linux. However, you can choose to use a different development environment as long as you produce standards compliant C++ code that builds and runs on the test platform.

Windows development environment

Instructions for working in Windows with Visual Studio and submitting your code to GitHub (and get feedback by email even before the final deadline) are posted here. (Note that there are other ways of linking your development environment with GitHub using a GUI: SourceTree, GitHub Desktop)

You should use only standard-compliant C++11 code. It is your responsibility for ensuring that your code will work in our testing environment. You can still test your code in a Linux environment: when you sync your code with GitHub it will be automatically built and run and the results emailed to you. See section below on “Automated testing.”

Linux development environment

Instructions for working in Linux and submitting your code to get feedback by email (even before the final deadline) are posted here.

Automated testing

We are piloting the use of a continuous integration web service that automatically tests your code and gives real-time feedback. This service is provided courtesy of Travis CI. The Travis service will wait for you to push code to GitHub. After you do, it will try to compile and build your code. It will then send you an email describing the outcome. The email also has a link to a dashboard web page you can view to see a detailed log of what worked, or didn’t. This way you can get immediate objective feedback about how well your code is
working. We recommend pushing your code to GitHub every time you reach a milestone and would like feedback on your progress. You can check the results of automatic testing on Travis at https://travis-ci.com/ (same login as your github account).

Grading rubric

Your grade will be comprised of two parts, Form and Function. Function refers to whether your code works properly as tested by the main function (80%). Form refers to the design, organization, and presentation of your code. An instructor will read your code and evaluate these aspects of your submission (20%).

Submission Deadlines

The project will be submitted in two stages.

Stage 1 (Design):
The goal in this stage is to form a team, study the problem, and come up with a design to solve the problem. Your design should include:

- What are the member variables for the classes (Student and Registrar) and their types
- A drawing of the relation between the objects of the two classes
- Pseudocode (i.e an informal English outline of the processing steps) of all the member functions

Include the above information in a single PDF document and upload to Titanium. Include the names of the team members and their sections. You are encouraged to discuss your design with your instructor.

Stage 2 (C++ Implementation):
You will be graded based on what you have pushed to the main branch of your GitHub repository as of the deadline. Commits made after the deadline will not be considered. Late submissions will not be accepted. Your code must compile/build for it to be tested and graded. If you only complete part of the project, make sure that it compiles before submitting.

Examinations

The midterm and final are not cumulative. Exams are closed book.

Class participation

You can show engagement in the course in many ways. These include attendance of the lectures, asking/answering questions in class, participating in class activities, attending office hours, and emails to the instructor.

Development Tool Resources

Students will have to write C++ programs for the projects. The following choices for a code development environment are available:
• **GNU/Linux with gcc:** A Debian-based GNU/Linux OS virtual machine ready for students use and Debian-style installation scripts are posted online at «https://gamble.ecs.fullerton.edu/resources/».

• **Microsoft® development tools:** Students may request a Dreamspark account at «http://dsreqform.ecs.fullerton.edu/». A student may then download, for free, full featured versions of Microsoft Visual Studio.

• **Apple® development tools:** Students can freely download Xcode through the App Store application bundled with OS X. Students may download Xcode directly from «https://developer.apple.com/xcode/».

**Remote Linux shell:** A CentOS-based shell server is available through secure shell (ssh) and secure file transfer protocol (sftp). The hostname is ecs.fullerton.edu. If your email address is malcolm@csu.fullerton.edu, then your username is ACAD\malcolm. If you are using a command-line ssh client, then your command to connect to ecs.fullerton.edu will be `ssh 'ACAD\malcolm@ecs.fullerton.edu'`. Your password is the same password as your CSUF Portal password.

**Supplemental Instruction (SI)**

Supplemental Instruction (SI) study sessions are offered for this course. SI sessions meet two to three times a week, throughout the semester. Supplemental Instruction is an academic assistance program which provides peer-led group study sessions to assist students in traditionally difficult courses.

SI sessions are led by a SI leader who has already mastered the course material and has been trained to facilitate group sessions where students can meet to improve their understanding of course material, review and discuss important concepts, develop study strategies and prepare for exams. SI is for everyone, and open to all students enrolled in this class; not just those students who are struggling. Attendance at SI sessions is free and voluntary. Students, who attend SI sessions weekly, typically earn higher final course and exam grades than students who do not participate in SI. Please bring your lecture notes, books, and questions with you.

You may attend any section’s SI sessions but preferably a session associated with your instructor.

SI sessions for this class will meet in room CS 209. The schedule is:

<table>
<thead>
<tr>
<th>Section</th>
<th>Lecture time</th>
<th>Instructor</th>
<th>SI Leader</th>
<th>SI Session time</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-DIS</td>
<td>TuTh 1:00PM - 2:15PM</td>
<td>Wenlin Han</td>
<td>Arshya Sharifian</td>
<td>T/Th 7:15-8:30PM</td>
</tr>
<tr>
<td>02-DIS</td>
<td>TuTh 5:30PM - 6:45PM</td>
<td>Anand Panangadan</td>
<td>Paul Miller</td>
<td>Tu/Thu 4:00-5:15PM</td>
</tr>
<tr>
<td>03-DIS</td>
<td>MoWe 4:00PM - 5:15PM</td>
<td>Doina Bein</td>
<td>Emily Bui</td>
<td>M/W 1:00 - 2:15 PM</td>
</tr>
<tr>
<td>04-DIS</td>
<td>TuTh 11:30AM - 12:45PM</td>
<td>Allen Holliday</td>
<td>Roy Redman</td>
<td>T/Th/F 10:30-11:20AM</td>
</tr>
<tr>
<td>05-DIS</td>
<td>MoWe 1:00PM - 2:15PM</td>
<td>Robert Kretschmar</td>
<td>Steve Sanchez</td>
<td>M/W 4:00PM - 5:15PM</td>
</tr>
</tbody>
</table>
Attendance Policy

Attending lectures is mandatory. Students are responsible for all course material and announcements regardless of whether they are present or absent. Attendance will not be recorded after the first class, and does not factor directly into grades.

Missing class as part of a documented accommodation is guaranteed to be excused. The ADA accommodated student must make a reasonable effort to coordinate any absences with the instructor.

Administrative drops

Any student who misses the first class meeting may be dropped from the class, unless they contact the instructor or Computer Science department within 24 hours.

Make Up Policy

The following kinds of projects cannot be evaluated, and will be assigned a zero score:

- Late submissions.
- Email submissions.
- Source code that cannot be compiled successfully.
- Input/output that is falsified or does not match the submitted source code.
- Submissions that are plagiarized or otherwise violate the collaboration guidelines.

Collaboration

Collaboration is not allowed on any exam or reading assignments. You may work on homework and projects in groups of 1-3. You may work with students from other sections of this course. If you work in a group, make one submission with all group members’ names. If you work in a group, make one submission with all group members’ names. You may work freely with your fellow group members, but must limit the input you get from sources outside your group:

- You may help each other understand the assignment and brainstorm general solutions, but each group must develop and submit their own distinct work.
- You may give each other technical support, for instance troubleshooting installing Visual Studio or logging in to TITANium.
- You must separate to develop your own detailed solution to the problem, and type in your own source code and project report.
Given these requirements, any submissions with identical excerpts, or excerpts that are identical up to superficial rearrangements, will be considered highly suspect of plagiarism.

Communication

You have a CSUF-supplied email account, and that is the only way I have of reaching you outside class. Check that account daily for important class announcements and individual messages. I try to respond to all emails within two working days, but occasionally may take longer than that. Plan accordingly, especially around deadlines.

Participation

In the context of this course, participation is defined as the following:

- Arriving to class prepared and on time.
- Taking notes.
- Actively listening to the lecture and asking questions when appropriate.
- Annotating code listings and handouts.
- Bringing any required materials to class.
- When needed/desired, seeking assistance to complete assignments.
- Barring an emergency, not leaving the class session early unless the instructor consents.
- Not distracting oneself or others with smartphones, games, online diversions, etc.
- Respecting and treating the instructor and the student’s peers civilly.

Required Material

- A writing instrument
- A notebook
- A personal computer with the requisite development tools or regular access to a computer lab

Academic Dishonesty

By submitting work for evaluation, the student acknowledges that he/she has adhered to the spirit of the university’s academic honesty policy and that his/her submission is an original work done by the student unless otherwise directed to work in groups. You are responsible for being aware of and following the spirit of CSU Fullerton’s academic honesty policy found at [http://www.fullerton.edu/senate/publications_policies_resolutions/ups/UPS%20300/UPS%20300.021.pdf](http://www.fullerton.edu/senate/publications_policies_resolutions/ups/UPS%20300/UPS%20300.021.pdf). Academic dishonesty includes such things as cheating, inventing false information or citations, plagiarism, and helping someone else commit an act of academic dishonesty. It usually involves an attempt by a student to show a possession of a level of knowledge or skill, which he/she in fact does not possess.

Cheating is defined as the act of obtaining or attempting to obtain credit for work by the use of any dishonest, deceptive, fraudulent or unauthorized means. Examples of cheating include, but are not limited to
using notes or aids or help of other students on tests and examinations in the ways other than those expressly permitted by the instructor, plagiarism as defined below, tampering with grading procedure, and collaborating with others on any assignment where such collaboration is expressly forbidden by the instructor. Plagiarism is defined as the act of taking the specific substance of another and offering it as one's own without giving credit to the source (e.g., copying other person's program).

When you use sources, you must acknowledge the original author or source following standard scholarly practice. **You are not allowed to any material from any website that provide solutions to the assignments given in class for a fee or free of charge.**

Failure to follow the spirit of the academic honesty policy will result in a severely negative evaluation of your work in question. Each offense will be reported to the Department Chair and to the Dean of Students office, Student Conduct. A first offense will result in a zero score on the offending assignment. A subsequent offense will result in an F in the course.

**ADA Accommodations**

Any student who, because of a disability, may require special arrangements in order to meet course requirements must register with the Office of Disability Support Services within the first week of classes. The Office of Disability Support Services’ website is «[http://www.fullerton.edu/DSS/](http://www.fullerton.edu/DSS/)». They can be reached by phone at 657-278-3117 or TDD at 657-278-2786. Their email address is «dsservices@fullerton.edu». Their office is located in University Hall, room 101. The instructor may request verification of need from the Dean of Students Office. Students requesting accommodations shall inform their instructors during the first week of classes about any disability or special needs that may require specific arrangements/accommodations related to attending class sessions, completing course assignments, writing papers or quizzes, tests or examinations.

**Emergency Procedures**

For your own safety and the safety of others, each student is expected to read and understand the guidelines published at «[http://prepare.fullerton.edu/campuspreparedness/](http://prepare.fullerton.edu/campuspreparedness/)». Should an emergency occur, follow the instructions given to you by faculty, staff, and public safety officials. An emergency information recording is available by calling the Campus Operation and Emergency Closure line at 657-278-4444.

**Instructional Continuity**

Due to an event such as an epidemic or a natural disaster that disrupts normal campus operations, students must monitor the course Titanium site and their campus email address for any instructions and assignments that the instructor announces.

**Extra Credit**

There are no opportunities for extra credit. Please do not ask for extra credit.
Recording & Transcription of Class Content

Recording class content is governed by UPS 330.230, «http://www.fullerton.edu/senate/documents/PDF/300/UPS330-230.pdf». Each instructor must permit class content to be recorded or transcribed by students when mandated to do so by the Americans with Disabilities Act or by other federal or state laws. Any recording of class content is for private use and study and shall not be made publicly accessible without the written consent of the instructor and students in the class.

Course Rules & Classroom Management

Unless an agreement or accommodation is reached between the student and the instructor, these rules must be followed.

- Attendance at all regularly scheduled lecture and discussion section is mandatory.
- Do not eat during lecture.
- If it makes noise, silence it.
- Portable computer use is not allowed in lecture except for taking notes.
- The student is responsible to be aware of any course announcements including changes to due dates and requirements.
- Homework, programming assignments, etc. may not be submitted late.
- Third party work (code, artwork, etc.) may not be used in student work without prior instructor consent. Failure to gain and document instructor consent will be construed as willful academic dishonesty.
- When a third party’s work is incorporated into student work after gaining instructor consent, failure to wholly document the work’s origin, copyright and license will be construed as willful academic dishonesty.

Acknowledgment

Portions of this syllabus draw from syllabi authored by Professors David Falconer, Matt Huffman, David A. Mix-Barrington, Mariko Molodowitch, Michael Shafae, and Kevin Wortman.