COP 3530 Syllabus

Data Structures & Algorithms
Fall, 2014

Catalog Description

Algorithm development using pseudo languages, basic program structures, program design techniques, storage and manipulation of basic data structures like arrays, stacks, queues, sorting and searching and string processing. Linked linear lists. Trees and multilinked structures. (M)

Course Overview

This course is an intense experience meant to transform the *novice* programmer into an *apprentice* programmer. The diligent and industrious student shall leave the course equipped with the understanding & fundamental tools necessary for serious programming. Students are responsible for keeping up with assigned readings and completing homework and programming assignments. This course is, and for good reason, the final gateway on the path to the advanced courses (e.g., Operating Systems, Network Fundamentals, and a whole host of cool technical electives)—make the most of it so that you are prepared for what awaits you.

Topics

Data and memory

- Machine level representation of data: bits, bytes, and words; binary representation of integers; representation of character data; representation of records and arrays

Data structures & abstract data types

- Fundamental data structures: linked structures; implementation strategies for stacks, queues, hash tables, graphs, and trees; representations of graphs; depth- and breadth-first traversals; strategies for choosing data structures

Algorithms, classification, and analysis

- Review of basic algorithmic analysis: asymptotic analysis of upper and average complexity bounds; identifying differences among best, average, and worst case behaviors; big "O," little "o," omega, and theta notation; standard complexity classes; empirical measurements of performance; time and space tradeoffs in algorithms; using recurrence relations to analyze recursive algorithms
- Fundamental computing algorithms: O(N log N) sorting algorithms (Quicksort, heapsort, mergesort); hashing, including collision-avoidance strategies; binary search trees
- Algorithmic strategies: brute-force algorithms; greedy algorithms; divide-and-conquer; backtracking; branch-and-bound; dynamic programming; heuristics; pattern matching and string/text algorithms; numerical approximation algorithms

Software development

- Review fundamental object-oriented programming: Object-oriented design; encapsulation and information hiding; classes; separation of behavior and implementation; class hierarchies; inheritance; polymorphism
- Software engineering: Software validation; testing fundamentals, including test plan creation and test case generation; object-oriented testing

You will have ample opportunity to practice and improve their programming skills. Specific, task-oriented techniques will be covered—but you are expected to be proficieny in programming using C++11. While we shall use C++ as our languages of discourse, though the principles covered shall transcend any particular programming language.

Relationship of course to program outcomes (ABET)

a) an ability to apply knowledge of mathematics, science, and engineering.
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.

Instructional staff

Name	**E-mail**	**Office**	**Office Hours**
Dave Small	dts@cise.ufl.edu	CSE E422	tba

Administrivia

Homepage

http://www.cise.ufl.edu/~dts/cop3530

Prerequisites

COP 3504 or COP 3503 with minimum grade of C, COT 3100, and MAC 2234, MAC 2312, MAC 3473 or MAC 3512.

Computer requirement

Access to and on-going use of a computer on which you can install software. *A laptop is strongly recommended (so you can bring it to class and office hours). Mac, Linux, PC… it's all good.*

Meeting times and places

See here

Required textbooks

Algorithms in C++, Parts 1-5: Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms, 3/E
by Robert Sedgewick
ISBN: 0201350882 — Parts 1-4 (Volume 1)
ISBN: 0201361183 — Part 5 (Volume 2)
ISBN: 020172684X — Parts 1-5 (Volumes 1 & 2 bundled)
ISBN: 0321735374 — Parts 1-5 & *C++ Backpack Reference Guide* (The "super bundle")

Recommended C++ reference book

C++ Primer, 5th edition
by Stanley B. Lippman, Josée Lajoie, Barbara E. Moo
(Addison-Wesley, 2012)
ISBN: 0-321-71411-3

Exam schedule — TENTATIVE

Exam 1: Monday, 9/29
Exam 2: Monday, 10/27
Exam 3: Monday, 11/24

Last day of class

Wednesday, 12/10

Philosophy and policy

Read the Course Policies page!

Final grades

I do not give grades: students *earn* their grades. Your final grade will be based on the number of grade points you have accumulated at the end of the semester. The grading scale will be no harsher than 100-92 = A, 91-89 = A-, 88-86 = B+, 85-82 = B, 81-79 = B-, 78-76 = C+, 76-72 = C, 71-69 = C-, 68-66 = D+, 65-62 = D, 61-59 = D-, 58-0 = E. During the semester you will accumulate points in categories weighted as follows:

10%	exam 1

20%	exam 2
30%	exam 3
30%	projects
5%	quizzes
5%	exercises (ungraded)

It is your responsibility to earn the grade you desire.

Projects

These are substantial programming challenges that will, for most students, require several days of effort to complete.

Exercises (ungraded)

Expect (ungraded) homework to be assigned every lecture. You are expected to do it to the best of your ability. Five percent of your grade will be based on the number of exercises you submit. For each assignment, you may submit either:

1. a completed assignment that is, to the best of your knowledge, perfect (i.e., 100% correct), *or*
2. an assignment is less than perfect *along with* a detail explanation of what is missing/incorrect and how you suspect it should be completed/fixed, *or*
3. a page that contains:
    - in the upper right-hand corner, mechanically **printed** (use a laser, inkjet, or other printer):
        1. your name
        2. COP 3530
        3. the date the homework will be collected
        4. the homework number
        5. the *period* and *room* in which your lab meets
    - in the center, *neatly* hand-written (in block print) the statement:

> Even though I didn't do it, I would still like credit for work whose

> purpose is to help me build the skills I will need to succeed in this and future programming-oriented courses. I promise not to complain should I do poorly on the quizzes or exams.

followed by your signature.

Submissions lacking the elements specified above will be ignored.

Mandatory assignments

Completion of the ungraded biosketch and postmortem assignments is *required* to pass the course. *Be sure to save all the work you do through out the semester as you'll need it for one of the postmortem deliverables.*

Academic dishonesty

ACADEMIC DISHONESTY WILL NOT BE TOLERATED. Unless otherwise explicitly stated, assignments are to be done individually. You are expected to do your own work: the point of an assignment is *not* the correctness of your submitted answer. *The one and only reason I assign homework is for you to learn by going through the process of solving a problem on your own.* I don't want someone else's solution, I want *your* best solution—as perfect or imperfect as it may be.

Individuals who misrepresent work as being their own, submit fabricated data, or otherwise engage in anti-intellectual behavior will be dealt with severely and reported to the Office for Student Judicial Affairs. For more information, consult the Dean of Students Student Conduct & Honor Code site.

Permitted activities

You may freely use any code and algorithms presented in the *required* textbook(s), provided by your instructor, or authored solely by yourself*. You may also use code and algorithms from any other source with *written* permission from the instructor.

*—when it comes to (re)using things that were not created specifically for this course, "authored solely by yourself" means just that: things developed using prohibited resources are tainted and thus may not be (re)used.

Prohibited activities

Everything that is not *explicitly* permitted is prohibited. For example, you may not:

- reference, use, or adapt solutions found online,
- reference, use, or adapt solutions found in other books,
- reference, use, or adapt solutions developed by past students,
- reference, use, or adapt solutions developed by other people,
- solicit someone to do all or part of the assigned work,
- provide your work to another student,
- &c.

The preceding list is not meant to be exhaustive. In short, if you have even the slightest doubt about whether something is permitted, error on the side of caution and *don't do it!*—as it's almost certainly not allowed. If you have any questions, ask. For more information, consult the Dean of Students Academic Honesty - Student Guide.

Example 1

Consider the case of an assignment where try as you might, you are unsuccessful in figuring out out how to do **X**.

Unacceptable behavior

Having completed the assignment to the best of your ability, you [prohibited activity] and now understand how to do **X**. Happy that you now understand what had previously eluded you, you then prepare and submit a solution that incorporates that new understanding.

ACCEPTABLE BEHAVIOR

Having completed the *entire* assignment to the best of your ability, you [prohibited activity] and now understand how to do **X**. Happy that you now understand what had previously eluded you, you prepare and submit the final deliverables *without* making any changes or additions to the work you had previously done.

Optional, but **strongly recommended**: for your own private amusement and edification, prepare a solution *from scratch* that incorporates that new understanding, but *without [prohibited activity] while you are creating it*—do *not* submit this solution! If, and only if, you do this, you *may* reference, use, or adapt the *solution you prepared*—and *only* your solution—when working on a future assignment in this course. You still are still not allowed to engage in any [prohibited activity] while working on the future assignment (for example, you aren't allowed to revisit the website that lead to the conceptual breakthrough which enabled you to formulate your unsubmitted solution).

EXAMPLE 2

You've decided to solve an exercise using an **X** and decide to use an implementation you wrote for some other course/personal project/&c. If you engaged in any [prohibited activity] when developing your **X** implementation, then it is tainted: therefore you are not permitted to reference, use, or adapt it (in any way) for assignments made in this course.

If you are certain that no [prohibited activity] was involved when developing the **X** implementation, you *may* use it for assignments in this course. You are, however, **strongly encouraged** to reimplement it from scratch—this will not only help reinforce the concepts, but by this point you probably have more programming experience and thus will likely devise a more elegant implementation.

Concluding remarks on academic honesty

As you can see from the examples, I want you to be intellectually curious and I want you to explore and learn independently. That, however, is not a license to circumvent the prohibited activities. I want the work you turn in to be your own. There is a very clear line: do not cross it.

Comments

My goal is simple: to help you learn — both inside and outside the classroom. If you have questions, there is no excuse for not getting help. The TAs, consultants, and I all hold office hours just for the purpose of helping you, either one-on-one or in small groups. No matter how busy we may look, during office hours, you have priority over everything else. If you have a problem or question, come by and we'll talk about it — don't put it off.

Modifications

This document is subject to revision as needed. All modifications will be noted in this section.

Things the college/university wants stated verbatim

19. This statement must be included in every grade scale on syllabi: “A C- will not be a qualifying grade for critical tracking courses. In order to graduate, students must have an overall GPA and an upper-division GPA of 2.0 or better (C or better). Note: a C- average is equivalent to a GPA of 1.67, and therefore, it does not satisfy this graduation requirement. For more information on grades and grading policies, please visit: http://www.registrar.ufl.edu/catalog/policies/regulationgrades.html”

20. Honesty Policy – All students admitted to the University of Florida have signed a statement of academic honesty committing themselves to be honest in all academic work and understanding that failure to comply with this commitment will result in disciplinary action. This statement is a reminder to uphold your obligation as a UF student and to be honest in all work submitted and exams taken in this course and all others.

21. Accommodation for Students with Disabilities – Students Requesting classroom accommodation must first register with the Dean of Students Office. That office will provide the student with documentation that he/she must provide to the course instructor when requesting accommodation.

22. UF Counseling Services – Resources are available on-campus for students having personal problems or lacking clear career and academic goals. The resources include:

- University Counseling Center, 301 Peabody Hall, 392-1575, Personal and Career Counseling.
- SHCC mental Health, Student Health Care Center, 392-1171, Personal and Counseling.
- Center for Sexual Assault/Abuse Recovery and Education (CARE), Student Health Care Center, 392-1161, sexual assault counseling.
- Career Resource Center, Reitz Union, 392-1601, career development assistance and counseling.

23. Software Use – All faculty, staff and student of the University are required and expected to obey the laws and legal agreements governing software use. Failure to do so can lead to monetary damages and/or criminal penalties for the individual violator. Because such violations are also against University policies and rules, disciplinary action will be taken as appropriate. We, the members of the University of Florida community, pledge to uphold ourselves and our peers to the highest standards of honesty and integrity.

Note: Statements in items 19-23, should be included as is.

[And so they were. Kinda. I did translate them from plain text to HTML compliant markup. Oh noes... I hear the UF syllabus police coming for me. Gotta run. K, thnx bai!—dts]