COP 3502C Section 1
Computer Science I - Summer 2020 Syllabus

Course Prerequisites: COP 3223 (Introduction to C Programming)
Class Time: Monday, Wednesday 4 – 5:50pm
Class Location: Online =)
Course Web Page: **http://www.cs.ucf.edu/courses/cop3502/sum2020**

Lecturer: Arup Guha
Office: My House =) **Email:** dmarino@cs.ucf.edu
Phone Number: 321-663-7749 (cell)
Office Hours: TR 10 am - 12 pm, or by appointment
7/13 – 7/24 by appointment only

I do NOT check my WebCourses email. Please email me at dmarino@cs.ucf.edu to contact me.

Teaching Assistants and Office Hours

Name	Email	Office Hours	OH Location
Jordan Collins	jecollins2019@knights.ucf.edu	F 9 am - 12 pm	Online
Daniel Foster	danielfoster@knights.ucf.edu	R 4 - 6 pm	Online
Alec Kerrigan	aleckerrigan@knights.ucf.edu	W 2 - 4 pm	Online
James Simmons	jamessimmons@knights.ucf.edu	R 12 - 2 pm	Online

Course Objectives

1. Introduce known algorithms.
2. Provide software skills in C.
3. Introduce mathematical tools necessary to analyze algorithms.
4. Introduce the problem solving technique of recursion.
5. Introduce implementing data structures from primitive constructs in C.
6. Introduce search and sorting techniques.

Reference Books: Any book on data structures will do for this course. The following book is available in the bookstore: Data structures, algorithms & software principles in C, Thomas B. Standish, Addison – Wesley (ISBN – 0-201-59118-9). There are a detailed set of course notes and sample programs that should be sufficient in explaining the material to most students.

Most Critical Course Items (due to Online Course Delivery)

1. It is much more tempting to partake in academic misconduct in an online course. If significant academic misconduct is detected, serious measures will be taken which make portions of the syllabus null and void. Last semester (Spring 2020), I threw out an entire exam due to academic misconduct and assigned an extremely low score to 93 students who engaged in academic misconduct. This semester, if I detect academic misconduct, I am liable to fail a student and report them to UCF. (I did not do this last semester, but need to be up front about my intentions now.) Please do not be tempted. If you are having difficulty, please contact me directly (cell phone or email) or speak to teaching assistants in the course.

2. In response to what occurred in one of my courses last semester, I am instituting some measures to prevent the wide-spread academic misconduct that previously occurred. Since I've never tried these techniques in my 20+ years of teaching, I am not sure how they will go. I may be forced to abandon some of all of them due to logistics. I am keeping an open mind with the goal of delivering a class equivalent in value to an in person version. Please watch ALL recorded videos (either live or soon afterwards) so that you can keep up with any changes I decide upon.

3. COP 3502 is a very challenging class. The average student should expect to spend 12 hours a week on the course. Just because the class is online, I don't want to compromise my students' education by watering it down. The upside to the course being challenging is that if you can get through this course with mastery of the skills taught in it, you are virtually guaranteed to graduate with a computer science (CS) degree from UCF, since mastering the course material will give you all the necessary skills in terms of diligence and problem solving that you'll need to properly handle future CS courses. (If you are in another major, mastering the skills in this course will allow you to handle most future coding challenges you may face.) You don't earn those skills for free. You earn them by putting in a great deal of effort. Make sure to plan a schedule that allows for this time consistently.

4. All of my course materials content wise (notes, sample programs, program and exam solutions, etc.) **EXCEPT VIDEOS** will be posted online via my course web page at:

http://www.cs.ucf.edu/courses/cop3502/sum2020

At a minimum students should visit the course web page every Monday, Wednesday and Friday for added materials.

5. Webcourses will be used to handle the course lecture videos and office hours, as well as the portal to submit assignments and receive grades. At a minimum students should visit Webcourses every Monday, Wednesday and Friday for class and other announcements.

6. From July 13 - 24, I am running an intensive summer high school competitive programming camp. Due to the time demands of the camp, I will be unable to tape lectures during the scheduled class period. I will tape the corresponding lectures early and ask students to view them by the scheduled class time. Once the camp starts, I'll get a better feel for my availability.

7. On occasion, to take advantage of the online format, I may record extra videos separate to class time. For example, I may do a video where I code up an old assignment from start to finish, to show students my coding process. Also, I plan on taping a tutorial on debugging. From time to time, if I see any other need for an extra video that I can't fit in class time, I may post it. Alternatively, what I may do is pre-tape a portion of a class video and make the live portion of my class shorter. Students are expected to watch these extra videos and exam material may be pulled from these videos.

Tentative Grading Procedures

The final letter grade will be based upon the seven items listed below. **Plus/minus grades will be issued, when deemed appropriate.**

Item	Weight
Week 1 Programming Assignment	2%
Individual Programming Assignments (5)	25% total (5% for each)
Recitation Programs (choose 2 of 4)	8% total (4% for each of these two)
Recitation Quiz	10%
Homework Assignments (via recitation)	5% total (1%, 2%, 2%)
Exam #1	20%
Final Exam*	30%

Also, in order to pass the class you must earn at least a 40% on the final exam. I reserve the right to either enforce or not enforce this clause, based on the individual situation.

Rather than use a "strict" 90 – 100 grading scale, I adjust my grade lines to account for difficult exams. My webpage discusses this process in detail.

Note: This grading breakdown is subject to change. Any changes will be discussed in course videos and Webcourses announcements. As previously mentioned, in the past I made changes to the class syllabus based on class behavior. In some classes I changed the syllabus to include class attendance in the middle of the semester and as previously mentioned, I threw out an entire exam grade due to academic misconduct. In classes where I've had a vast majority of responsible students, I have not needed to make any changes to the class grading system.

Programming Assignments (Week 1 Assignment and Individual Assignments)

All programming assignments will be turned in over WebCourses. All programs must be done in C. The specific compiler used for grading will be discussed in the first class video. Programs must be done individually with course staff help only. **Collaboration is not allowed on any programming assignment**. (A further explanation of academic misconduct on programming assignments is discussed on my main UCF web page. Please read this.) **Official assignment due dates will be posted on WebCourses.** (Please look these up yourself instead of asking a friend.)

THE ONLY VALID DUE DATES ARE THOSE POSTED ON WEBCOURSES.

My personal advice is to submit all assignments **AT LEAST THREE HOURS BEFORE THE POSTED DEADLINE**. Too often, students wait till the last minute only to miss the deadline due to network issues. **IN CASES WHERE A SUBMISSION IS LATE (EVEN BY A SECOND), A GRADE OF ZERO WILL BE GIVEN TO THE SUBMISSION.**

Recitation, Homework Assignments, Quiz

On some weeks of recitation, a programming problem will be given. Each of these is on the website open.kattis.com, but students may be required to turn in more than their source code. Directions will be given in recitation. Students are expected to submit two out of four of these via Webcourses for credit. (The lowest two grades will be dropped.) On other weeks of recitation, either a written homework assignment will be introduced, an exam review will be given, or a quiz will be given. For homework assignments, the goal will be to use the recitation time to get started on these questions and ask questions of the teaching assistant when necessary. The homework assignments will be submitted via Webcourses at a later time. Note that some of these questions may well be old Foundation Exam questions with solutions online, but that students will gain more benefit from working through these problems fully on their own. Students may work together to some extent on these homework questions with the understanding that the goal is to prepare themselves for exams, and for many students, the Foundation Exam. Note that all the programming assignments must be done individually and help can only be obtained from the course staff or other staff assigned to help for the course (SARC). Exam reviews will be simply extra help for students to prepare for exams. They'll typically involve looking at past exam questions, giving students some time to think about them, and then having the TA go over various approaches to solve the problems. One week of recitation will actually be a quiz. The quiz is mostly a test run for the exams with the goal of giving students early feedback on how they are doing and also for the course staff to assess the exam plans.

Exams

Students will be allowed to use their course notes for exams as well as a calculator for **basic** functions, but students should NOT communicate with anyone directly or indirectly or access any webpages except Webcourses during the exam. ***In addition, grading criteria may be created to only award full credit to solutions that could have been devised without a calculator.*** To limit the chance of cheating, both exams will be conducted where students will have a short amount of time to submit answers to one or two questions, multiple times. So, each exam will be multiple windows of 15 or 20 minute time periods.

In addition, if some sort of academic misconduct is suspected, I (Arup Guha, course instructor), reserve the right to conduct a live Zoom technical interview. If the results of the interview aren't consistent with the exam score, I reserve the right to throw the exam score out and score the interview and use it as the exam replacement grade.

Tentative Schedule

Week	**Monday Class**	**Wednesday Class**	**Recitation**
May 11 – 15	Course Intro	Dyn Mem Alloc	Kattis Introduction Rec Prog #1 (alice)
May 18 - 22	Dyn Mem Alloc	Linked List 1	Homework #1
May 26 – 29	**No Class**	Linked List 2	**Quiz**
June 1 – 5	Stacks	Queues	Exam 1 Review
June 8 – 12	**Exam #1**	Alg Analysis	Homework #2
June 15 – 19	Recursion 1	Recursion 2	Rec Prog #2 (stars)
June 22 – 26	Sorting	Binary Trees 1	Rec Prog #3 (no win)
June 29 – July 3	Binary Trees 2	AVL Trees	**Cancelled**
July 6 – 10	Tries	Binary Heaps	Rec Prog #4 (bool tree)
July 13 – 17	Bit Ops, Hash Tables	Backtracking	Homework #3
July 20 – 24	Binary Search Apps	F. E., Prog Team	Final Exam Review
July 27 – 31	Final Exam Review	**Final Exam**	**Cancelled**

- **This schedule is tentative. The only items guaranteed to be on the listed dates are the exams. All other items may be shifted based on how the class actually runs. These details will generally only be discussed in the videos and Webcourses announcements.**

- **Note: All program and recitation program due date/times will ONLY be posted on Webcourses. Please go there to find when each of these assignments is due.**