COP 3223 Syllabus
Introduction to C Programming – Fall 2025
Section 1

Course Prerequisite: COP 2500 or Placement Test
Class Time: MWF 10:30 – 11:20 am
Class Location: CB1-104
Course Web Page: http://www.cs.ucf.edu/courses/cop3223/fall2025

Instructor: Arup Guha
Office: HEC – 240 **Email:** dmarino@ucf.edu
Phone Number: 407-823-1062
Office Hours: Listed on course webpage

Course Description: COP 3223 provides an introduction to programming in C for those with ***the equivalent of one prior class of programming experience***.

We will cover the following topics in C: input, output, variables, arithmetic expressions, if statements, loops, loop control, nested loops, functions, use of pointers to pass parameters, one dimensional and two dimensional arrays, strings, arrays of strings, structs and linked lists. Depending on how well the class does, other concepts may be introduced.

Recommended Book(s): Any book that covers the basics of C will do. Many students who succeed in the course never buy a book because online resources for both languages are excellent. If you prefer to have a book, please feel free to ask me if a particular book would be suitable as a supplement for our course. The reality is that when a course textbook existed, reading it did not correlate with the course grade **at all.** **Writing extra programs did though.**

Other Important Course Policies

1) Cheating will not be tolerated. **If a student is caught cheating, then the grade on that assignment for all students knowingly involved (the person providing answers as well as the one taking the answers) will be a -25%.** **(Note, this is less than 0%.)** Since discussion of concepts with other students is often helpful, cheating must be more clearly defined. In particular, the following items are cheating: **getting any code directly from any generative AI, even if it's modified afterwards**, copying a segment of code of three lines or more from another student from a printout or by looking at their computer screen, taking a copy of another student's work and then editing that copy, and sitting side by side while writing code for assignments and working together on segments of code. In all of these situations, **BOTH people responsible**, the one from whom the three lines of code are taken as well as the person who takes those lines of code are engaging in academic misconduct. For example, if someone makes an electronic copy of their code accessible to ANYONE in the class (except for themselves) before 48 hours after an assignment is due, they are automatically culpable of academic misconduct. It does not matter if the recipient of the code doesn't use it, uses it a little, or copies it directly. Furthermore, based on the severity of the case, the entire course grade for the student may be lowered an entire letter grade.

If you get stuck on an assignment, please ask either a TA or me for help instead of getting help from another student. Part of the learning process in programming involves debugging on your own. In our experience, when a student helps another student with an assignment, they rarely allow the student getting help to "figure out" problems on their own. Ultimately, this results in a lack of debugging experience for the student receiving help. The goal of the TAs and instructors is to provide the facilitation necessary for students to debug and fix their own programs rather than simply solving their problems.

You are encouraged to work together on the lab exercises and other ungraded exercises and use those as an avenue to improve your understanding of concepts.

2) In order to take a make-up quiz or exam, you must request one from the instructor. The instructor will grant requests using his own judgment by applying the following general rule: "Make-up exams will only be given if the reason for missing the exam was out of the student's control." For example, being hospitalized unexpectedly is out of a student's control, but oversleeping or going to happy hour is not out of a student's control. ***If possible, it is recommended that the instructor be contacted before the exam.***

3) NO LATE ASSIGNMENTS WILL BE ACCEPTED. If a work/military/family/personal reason prevents you from turning in an assignment on time, please contact me as early as possible and I will make alternate arrangements. If the situation is serious enough, I may suggest pursuing an Incomplete or dropping the course.

4) Both the course web page and WebCourses will be crucial elements of the course. ***It is your responsibility to check both of these before every class meeting for any updates that may be posted.*** Some clarifications may only be given in class and won't be posted online at all, so make sure you keep up with announcements in class.

Important Note

In addition to the programming assignments, a good way to learn is to copy and paste code examples from the course web page and edit them. Also, working on ungraded assignments in groups may help some students.

This class is far more time consuming than other introductory classes for many students. Unlike other introductory classes which simply ask students to read and regurgitate information, this class requires students to learn a new language, and then use that language to solve non-trivial problems that ***students have not seen before.*** It takes most students some time to get comfortable and confident with their programming skills. For each student, the amount of time and practice necessary to reach this level is different. Although we don't want to scare anyone, we do want to be up front about the fact that this class is a lot of work, and it's a different type of work than many other classes. (For example, it's possible you may get stuck on one error for a couple hours without making any progress. Once you fix that error, you might be able to finish your whole program in another thirty minutes.) ***I would imagine that for a good introductory student, on average, this class will require a total of 10 hours of work per week.***

Grading

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

Item	Percentage
Individual Homework Assignments (10 total)	20
Lab Participation	15
4 Quizzes (10% each)	40
Final Exam	25

In order to pass the class you must earn at least 40% on the final exam. (Thus, if you have 75% in the course but earn 30% on the final, you still get a C- in the course even though your percentage may qualify for a B.)

Letter Grades

I assign letter grades a bit differently than other professors. I do not use a straight 90-100, 80-90, etc. grading scale. Rather, at the end of the semester, I chose my lines for each grade. The drawback to this technique is that students do not know exactly what letter grade they are earning during the semester. The advantage to it is that if I make a difficult quiz/exam, I can adjust my grades accordingly so students don't get punished for my exam making skills. In the past, my A line has ranged from about 83-87%, my B line has ranged from 67-70%, and my C line typically ranges from 52-55%. I do not guarantee that these will be accurate for this semester, but I wanted to give you a rough idea of how the grades have gone in the past. After each exam, I will update WebCourses to show you what letter grade you have at that point in time. If you have further questions about my grading philosophy, please read the document I have posted at http://www.cs.ucf.edu/~dmarino/ucf. **Note: This grading breakdown is subject to change. Any changes will be discussed in class.**

Individual Programming Assignments

All programming assignments will be turned in over WebCourses. It's critically important to do these assignments in order to aid understanding of the course material. In order to grasp the material fully and feel comfortable programming in any language, one needs to write **MORE** programs than are assigned. Thus, while I'll have 10 graded programming assignments, it is expected that students will write or edit 2-3 programs a week **on top of** the assigned programming assignments, in order to get the proper practice. In addition, several practice exercises will be given in lab. Students must work on these prompts during lab and if they want to get more practice, can do so afterwards. This practice is especially important for beginning students. If there are any ambiguities in any program description, please just ask me for my intent. **The due dates for each assignment will be posted on WebCourses ONLY.**

Late Assignment Policy

NO LATE HOMEWORK ASSIGNMENTS WILL BE ACCEPTED. Due to possible server issues, it is **strongly suggested** that you attempt to submit programs **at least three hours** before the actual time it's due.

Labs

In each lab, a few small exercises will be given so students can get extra programming practice. The teaching assistants who run the labs will ensure that students are using the time in lab for the designated programming exercises (and not the individual class homework or work for other courses). Students are encouraged to get help as needed from both the TA and other students so that they understand the concepts better. 10% of the course grade will be based on TAs informally checking off work completed during lab while 5% of the course grade will be based solely on attending the lab. Please save **ALL** of the work you do in lab exercises so that if a TA asks to see your work for the last several weeks, you can show them.

Quizzes/Final Exam

For each exam, you'll be allowed to use a limited amount of notes as an aid. The specifics for the allowable aids will be given in the class period before each quiz/exam.

Community Service Opportunity

To get automatic full credit for 25 points (out of 125) on the Final Exam, you can do 5 hours of community service with a registered 501(c)(3) organization **BEFORE November 16, 2025**, and turn in the required signed form and activity summary (more details on the course web page) by the **10:30 am on November 17, 2025.** Note, you will only get full credit if I receive the signed form and write up by 10:30 am on that day. If I receive it at 10:40 am on November 17th, then you'll have to take the last portion of the Final Exam to earn points. Every semester, a couple students are just a couple minutes late and I don't count their forms. **Please do the community service early and submit the forms to me way in advance, so that this doesn't happen to you.**

AI Policy

Try to limit your use of AI while in the beginning learning phases. My experience has been that the more students rely on AI, the less they actually learn. A number of studies are now coming out showing the negative effects of using AI for the purposes of education. Here is a link to the summary of one such study: https://time.com/7295195/ai-chatgpt-google-learning-school/.

The problem with using AI early in the learning process (other than the fact that you turn your brain off) is that AI still makes mistakes and it writes code that's very difficult to debug. But to get to the point where (a) you can realize the AI is wrong, and (b) you can fix it, you have to learn all the basics from the ground up and fully understand how things work. (This is similar to how mathematics students aren't allowed to use calculators until high school. Or at least, they shouldn't be allowed to use them until high school.)

Course Assistance

You may attend the instructor's office hours or TA's office hours for general help understanding course material as well as help debugging individual programming assignments. Two learning assistants are assigned to the course. I will utilize them as follows: they will book one on one tutoring sessions each week with any student who requests one. For students with lower quiz grades, attending these sessions will add points to their quiz grades. Details will be discussed in class. Each LA can book up to 9 sessions a week.

Tentative Schedule

Week	Monday	Wednesday	Friday	Sunday
Aug 18-22	Variables, I/O	Formula Probs	Arith Expr	**P1 Due**
Aug 25-29	Calling Functions	Math, Random	Practice Probs	**P2 Due**
Sept 2-5	**Labor Day**	Quiz Rev (Zoom)	**Quiz #1**	
Sept 8-12	If Stmt	If Stmt	While Loop	**P3 Due**
Sept 15-19	For Loop	Loop Examples	Loop Control	**P4 Due**
Sept 22-26	Nested Loops	Quiz Review	**Quiz #2**	
Sept 29-Oct 3	C functions	C functions	C functions	**P5 Due**
Oct 6-10	Arrays	Arrays	Sorting	**P6 Due**
Oct 13-17	Array Examples	Quiz Review	**Quiz #3**	
Oct 20-24	2D Arrays	2D Arrays	Bingo Example	**P7 Due**
Oct 27-31	Char. Process	Strings	Arrays of Strings **WD Deadline**	**P8 Due**
Nov 3-7	Sorting Strings	String Examples	Structs	**P9 Due**
Nov 10-14	Struct Array	Struct Design	**Quiz #4**	
Nov 17-21	Linked lists	Linked lists	FE Review	**P 10 Due**
Nov 24-25	FE Review	**Thanksgiving**	**Thanksgiving**	
Dec 1-5	**Final Exam Dec 1 (10am-1pm)**			

Note: Assignments will be given in class and will be due over WebCourses. Tentative dates are given above for the assignments but consult WebCourses for the final due dates and times. Also, this schedule may change based on the pace of lectures.