

<b>Department:</b>	Computer Science
<b>Course Title:</b>	Introduction to Computer Science II
<b>Course #:</b>	CS32/187096910
<b>Credit Hours:</b>	4
<b>Lecture Time/Room:</b>	MW 10:00 – 11:50 A.M./Online
<b>Discussion Sections:</b>	1A: F 10:00 – 11:50 A.M./Online (Kareem Ahmed) 1B: F 12:00 – 1:50 P.M./Online (Xiheng (Lawrence) Xu) 1C: F 2:00 – 3:50 P.M./Online (Xuelu (Shirley) Chen)

I. Prerequisites

- a. Computer Science 31 with a D- or better

II. Course Materials

- a. Textbook: “Data Abstraction and Problem Solving with C++: Walls and Mirrors”, 7th ed., Frank M. Carrano and Timothy M. Henry, ISBN-13: 9780134463971, 2017, Addison Wesley
  - i. NOTE: The fourth, fifth, or sixth editions will suffice. Also, other good C++ books I could recommend are Accelerated C++, by Andrew Koenig and Barbara E. Moo, Addison Wesley, 2000, and Thinking in C++, Second Edition, by Bruce Eckel, Addison Wesley, 2000; the latter book is available online for free.
- b. Recommended Supplies: All students are encouraged to bring a USB flash drive to class to save class examples, your programming assignments, and notes on.

III. Course Description

- a. In this course, you will learn how to employ data abstraction to build programs larger than those you constructed in CS 31. You will learn how to use a variety of data structures in the course of solving problems, and importantly, how to analyze what data structures and algorithms are most appropriate for a given problem. In addition, you will deepen your understanding of modern programming by learning the principles of object-oriented and generic programming.
- b. Lectures will present material from the course textbook, supplemented by additional subject matter. You are responsible for **all** material presented in **all** lectures. All lectures will be done via Zoom, recorded, and posted on CCLE.
- c. In discussion sections, your TA will review material covered in the lectures, present subject matter not covered in the lectures, discuss homework and programming projects, and answer questions. All discussion sections will be done via Zoom, recorded, and posted on CCLE.

## IV. Course Evaluation

- a. Midterm Exams (2): 150 points each (300 points total)
- b. Final Exam: 150 points
  - i. NOTE: All exams are open-book, open-notes.
  - ii. The midterms will cover material from the lectures and the textbook. The final examination will focus on the material that's been covered since the second midterm exam, although there may be a few questions testing previous concepts. Missing an exam for any reason, without prior arrangements, will result in an exam score of zero.

## c. Assignments: 550 points

- i. Assignments will consist of 4 homework assignments and 4 projects. Here are the point values and the schedule of due dates:

<b><u>Assignment</u></b>	<b><u>Possible Points</u></b>	<b><u>Due Date</u></b>
Project 1	10	Friday, 7/2/2021
Homework 1	30	Friday, 7/9/2021
Project 2	140	Friday, 7/16/2020
Homework 2	30	Saturday, 7/24/2021
Project 3	140	Saturday, 7/31/2021
Homework 3	30	Sunday, 8/8/2021
Homework 4	30	Monday, 8/16/2021
Project 4	140	Friday, 8/20/2021

- ii. Assignments are posted on the class website and the syllabus and are due on the dates noted at 11:59:00 PM. Submissions are to be done via CCLE. NO late submissions will be accepted.
- iii. Before submitting your assignment, make sure your program can produce all the dialogue and output shown in the assignment. Every C++ program you turn in for this class must run correctly under two compilers: Visual C++ 2019 (PC) and clang++ or g++ (either Mac or Linux). The reason for this is to allow for proper sanity checking in multiple environments. If it does not run in both, your assignment will result in a zero. Your code will be tested using a script called g32, which factors in both styles of compilers.
- d. The course grade will be based on the percent of the total points (1000 points) earned as follows:

<b><u>%</u></b>	<b><u>Grade</u></b>
90 – 100	A
80 – 89	B
70 – 79	C
60 – 69	D
0 – 59	F

What will be different for this particular session is that a Pass/No Pass (P/NP) option is available if you wish to do so. Be careful in selecting this option, as doing so may affect your future plans in graduate school and/or employment potentially.

- e. The total points you earn from projects and homework (scaled to 0 through 100) will be capped at 30 points above the mean of your exam scores (scaled to 0 through 100). For example, if you average 90 on the assignments, your midterm score is 60, and your final is 40, then your assignment average is treated as only 80 (because that's 30 more than the mean of 60 and 40). In other words, your assignment scores won't count fully if you can't show from your exam scores that you learned what you should have from the assignments.
- f. A request for reconsideration of the grading for an item must be made within one week of our sending you your score for that item.
- g. You will not earn points on any of your coursework unless you have signed and turned in the Academic Integrity Agreement.

V. Course Plan for Success

- a. Read:
  - i. Each assignment comes with some written specifications, as well as class handouts and notes. The notes are more specific to each assignment. There may also be links to external references. All these resources are important. There are also lists of vocabulary words with the assignments. Learn these. Read each assignment very carefully and make certain you know the goal. If it seems too complicated, maybe your view of it is too complicated. When in doubt, ask questions.
- b. Program:
  - i. You cannot learn how to write large programs without writing large programs. There will be four large programming projects. Your program correctness score is based on your program's correctness as determined by our testing. The amount of time you spent working on the program is irrelevant; indeed, if you follow our software development advice, you'll probably spend less time and get a higher score than if you don't.
  - ii. Since the projects cannot give you experience with all the material in the course, there will be four homework assignments in addition to the projects. These may require you to answer some questions and analyze or write small programs. Some of the work you put into the homework will do double duty: it will help you with a programming project or will serve as a good study guide for an exam. If you are seriously interested in mastering the course material, you will do every homework problem, even though not

every problem of every homework assignment will be graded. (You won't know which problems won't be graded.)

c. Interact:

- i. Myself and the teaching assistants are here to help. The way we will be doing this is via Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TAs, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email [team@piazza.com](mailto:team@piazza.com). Find our class page at: <https://piazza.com/ucla/summer2021/cs32/home>. I will try to answer e-mail questions as quickly as possible, but Piazza will be the best way to communicate, as I will be checking that more frequently. Here is some netiquette to keep in mind:

1. NO YELLING, PLEASE

There's a time and a place for everything—BUT IN MOST SITUATIONS TYPING IN ALL CAPS IS INAPPROPRIATE. Most readers tend to perceive it as shouting and will have a hard time taking what you say seriously, no matter how intelligent your response may be. If you have vision issues, there are ways to adjust how text displays (Links to an external site.) so you can still see without coming across as angry.

2. Sarcasm can (and will) backfire

Sarcasm has been the source of plenty of misguided arguments online, as it can be incredibly difficult to understand the commenter's intent. What may seem like an obvious joke to you could come across as off-putting or rude to those who don't know you personally. As a rule of thumb, it's best to avoid sarcasm altogether in an online classroom. Instead, lean toward being polite and direct in the way you communicate to avoid these issues.

3. Don't abuse the chat box

Chat boxes are incorporated into many online classes as a place for students to share ideas and ask questions related to the lesson. It can be a helpful resource or a major distraction—it all depends on how well students know their classroom netiquette.

"Rather than asking relevant questions or giving clear answers, students might use the chat box to ask questions irrelevant to the discussion, or to talk about an unrelated topic," says Erin Lynch, senior educator at Test Innovators. The class chat box isn't an instant messenger like you'd use

with friends. Treat it like the learning tool it's meant to be, and try not to distract your classmates with off-topic discussions.

4. Attempt to find your own answer

If you're confused or stuck on an assignment, your first instinct may be to immediately ask a question. But before you ask, take the time to try to figure it out on your own. For questions related to class structure, such as due dates or policies, refer to your syllabus and course FAQ. Attempt to find the answers to any other questions on your own using a search engine. If your questions remain unanswered after a bit of effort, feel free to bring them up with your instructor.

5. Stop ... grammar time!

Always make an effort to use proper punctuation, spelling and grammar. Trying to decipher a string of misspelled words with erratic punctuation frustrates the reader and distracts from the point of your message.

On the other hand, it's important to be reasonable about others' grammar mistakes. Nobody likes the grammar police, and scolding a classmate because he or she used "your" instead of "you're" isn't practicing proper netiquette.

6. Set a respectful tone

"An increasingly common netiquette faux pas is treating e-correspondence with faculty and staff as an ongoing chat among friends," says Alexey Timbul, online professor at the Academy of Art University (Links to an external site.).

Every day may feel like casual Friday in an online classroom where you don't see anyone in person, but a certain level of formality is still expected in your communication with instructors. In addition to proper punctuation and spelling, it's good netiquette to use respectful greetings and signatures, full sentences and even the same old "please" and "thank you" you use in real life.

7. Submit files the right way

You won't be printing assignments and handing them to your teacher in person, so knowing how to properly submit your work online is key to your success as an online student. Online course instructors often establish ground rules for file assignment submissions, like naming conventions that help them keep things organized or acceptable file formats. Ignoring these instructions is a common example of bad netiquette.

“Receiving work that does not adhere to the file format and naming protocol means a student is not paying attention,” says Timbul. If you don’t follow instructions, you’re taking the risk that your instructor won’t be able to find or open your assignment. Save yourself and your instructor a headache and read their instructions carefully before submitting.

8. Read first

Take some time to read through each of the previous discussion post responses before writing your own response. If the original post asked a specific question, there’s a good chance someone has already answered it. Submitting an answer that is eerily similar to a classmate’s indicates to the instructor that you haven’t paid attention to the conversation thus far.

Remember, discussions can move fairly quickly so it’s important to absorb all of the information before crafting your reply. Building upon a classmate’s thought or attempting to add something new to the conversation will show your instructor you’ve been paying attention.

9. Think before you type

A passing comment spoken in class can be forgotten a few minutes later, but what you share in an online classroom is part of a permanent digital record. “Whether or not privacy settings are in place, the internet has a tendency to house things forever, and what you say privately can easily become public,” advises etiquette expert (Links to an external site.) Devoreaux Walton.

Not only is it good practice to be guarded when it comes to personal information, you always want to be just as respectful toward others as you would be if you were sitting in the same room together. Zink says a good rule of thumb to follow is if you’re comfortable standing up in front of a classroom and saying your message, then it’s most likely okay to share.

10. Be kind and professional

Online communication comes with a level of anonymity that doesn’t exist when you’re talking to someone face-to-face. Sometimes this leads people to behave rudely when they disagree with one another. Online students probably don’t have the complete anonymity that comes with using a screen name, but you could still fall prey to treating someone poorly because of the distance between screens. Make a

point to be kind and respectful in your comments—even if you disagree with someone.

“At the core, all of these mistakes come down to forgetting that an online classroom is still a classroom,” Lynch says.

“Good netiquette means conducting yourself in an online class with the same respect, politeness and professionalism that you would exhibit in a real-life classroom.”

- ii. Don't ask questions line-by-line as you type your program. Compile and debug programs on your own and find out what happens. When you try this and still don't understand, then send the program (via attachment) and the error message or whatever to indicate where the program doesn't work as expected.
- iii. I'm also available during office hours. A few minutes at the beginning can save hours toward the end.

## VI. Academic Integrity

At <http://www.deanofstudents.ucla.edu/Academic-Integrity>, the Office of the Dean of Students presents University policy on academic integrity, with special attention to cheating, plagiarism, and student discipline. The policy summaries don't specifically address programming assignments in detail, so we state our policy here.

Of course, you understand that your work on programming assignments must be your own. But we understand that high-level discussions about approaches to a problem have educational value and are acceptable. So where do we draw the line? We'll decide each case on its merits, but here are some categorizations:

### **Acceptable**

- Clarifying what an assignment is requiring
- Discussing algorithms for solving a problem, perhaps accompanied by pictures, without writing any code
- Helping someone find a minor problem with their code, provided that offering such assistance doesn't require examining more than a few lines of code
- Turning in someone's work without crediting the author of that work, if the source of that work is the course text or if that work was produced at or after the start of this quarter by a CS 32 instructor or a CS 32 TA

### **Unacceptable**

- Turning in any portion of someone's work without crediting the author of that work, if the source of that work is not the course text and if that work

was either not produced by a CS 32 instructor or a CS 32 TA or was produced before the start of the quarter

- Using or adapting a portion of or all of a solution from earlier offerings of this or any other class
- Soliciting help from an online source where not all potential respondents are subject to the UCLA Student Conduct Code
- Receiving from another person (other than a CS 32 instructor or a CS 32 TA) a code fragment that solves any portion of a programming assignment
- Writing for or with another student a code fragment that solves any portion of a programming assignment
- Helping the same person find problems with their code more than a few times for a particular assignment

Be especially careful about giving a copy of your work to a friend who "just wants to look at it to get some ideas". Frequently, that friend ends up panicking and simply copies your work, thus betraying you and putting you through the hassle of an academic discipline hearing. The Dean has additional advice.

You must abide by this policy in addition to the policies expressed in the UCLA Student Conduct Code. In accordance with University procedures, we will submit cases of suspected violations of this policy to the Dean. You will be signing a form agreeing to and understanding the UCLA Student Conduct Code through Google Forms at this link: [Academic Integrity Agreement](#)

## VII. Instructor and TA Info

<b>Instructor:</b> <b>Phone #:</b> <b>Office Hours:</b> <b>e-mail:</b> <b>Website:</b>	Professor Edwin Ambrosio (310) 592-2756 MW 8:45 – 9:45 A.M., Tu 3:30 – 4:30 P.M. (Online) <a href="mailto:eambrosi@g.ucla.edu">eambrosi@g.ucla.edu</a> <a href="https://ccle.ucla.edu/course/view/211A-COMSCI32-1">https://ccle.ucla.edu/course/view/211A-COMSCI32-1</a>
<b>Teaching Assistant:</b> <b>Office Hours:</b> <b>e-mail:</b>	Kareem Ahmed <a href="#">TBD</a> <a href="mailto:kareem.yousri@gmail.com">kareem.yousri@gmail.com</a>
<b>Teaching Assistant:</b> <b>Office Hours:</b> <b>e-mail:</b>	Xuelu (Shirley) Chen <a href="#">TBD</a> <a href="mailto:shirleychen@cs.ucla.edu">shirleychen@cs.ucla.edu</a>
<b>Teaching Assistant:</b> <b>Office Hours/Office:</b> <b>e-mail:</b>	Ziheng (Lawrence) Xu TBD <a href="mailto:lawrencexu@g.ucla.edu">lawrencexu@g.ucla.edu</a>



## Course Outline (subject to changes by the instructor)

<b><u>Week</u></b>	<b><u>Day</u></b>	<b><u>Chapters</u></b>	<b><u>Subject</u></b>
1	6/21/2021	1	Introduction/C++ Review
	6/23/2021	3	Pointers, Dynamic Arrays, Resource Management
2	6/28/2021	4, 8	Linked Lists
	6/30/2021	4, 8	Linked Lists (continued)
3	7/5/2021	N/A	NO SCHOOL – Independence Day
	7/7/2021	6, 7, 13, 14	Stacks and Queues
<b>4</b>	<b>7/12/2021</b>	<b>N/A</b>	<b>Midterm #1</b>
	7/14/2021	3	Advanced C++, Inheritance, Polymorphism
5	7/19/2021	2, 5	Recursion
	7/21/2021	2, 5	Recursion (continued)
6	7/26/2021	Interludes 6, 7	Templates, Iterators, STL
	7/28/2021	10, 11	Algorithmic Efficiency, Sorting
<b>7</b>	<b>8/2/2021</b>	<b>N/A</b>	<b>Midterm #2</b>
	8/4/2021	11, 15	Trees
8	8/9/2021	18	Hash Tables, Tree-Based Tables
	8/11/2021	17, 20	Heaps
9	8/16/2021	20	Graphs
	<b>8/18/2021</b>	<b>N/A</b>	<b>Final</b>