

Department:	Computer Science
Course Title:	Introduction to Computer Science II
Course #:	CS32/187096910
Credit Hours:	4
Lecture Time/Room:	MW 10:00 – 11:50 A.M./Online
Discussion Sections:	1A: F 10:00 – 11:50 A.M./Online (Steven Zeng) 1B: F 12:00 – 1:50 P.M./Online (Hengda Shi) 1C: F 2:00 – 3:50 P.M./Online (Seungbae Kim) 1D: F 10:00 – 11:50 A.M./Online (Kareem Ahmed)

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:

<u>Assignment</u>	<u>Possible Points</u>	<u>Due Date</u>
Project 1	10	Friday, 7/3/2020
Homework 1	30	Friday, 7/10/2020
Project 2	140	Friday, 7/17/2020
Homework 2	30	Saturday, 7/25/2020
Project 3	140	Saturday, 8/1/2020
Homework 3	30	Sunday, 8/9/2020
Homework 4	30	Monday, 8/17/2020
Project 4	140	Friday, 8/21/2020

- 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:

<u>%</u>	<u>Grade</u>
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. Find our class page at: <https://piazza.com/ucla/summer2020/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.
- 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: <https://forms.gle/X1G3NhbUxrc9XAh8A>

VII. Instructor and TA Info

Instructor:	Professor Edwin Ambrosio
Phone #:	(310) 592-2756
Office Hours:	M 8:50 – 9:50 A.M.; T 5 – 6 P.M.; W 12 – 1 P.M. (Online)
e-mail:	eambrosi@q.ucla.edu
Website:	https://ccle.ucla.edu/course/view/201A-COMSCI32-1
Teaching Assistant:	Kareem Ahmed
Office Hours:	Th 10 A.M. – 12 P.M.
e-mail:	kareem.yousrii@gmail.com
Teaching Assistant:	Seungbae Kim
Office Hours:	Th 1 – 3 P.M.
e-mail:	sbkim@cs.ucla.edu

Teaching Assistant: Office Hours/Office: e-mail:	Hengda Shi W 4 – 6 P.M. sihgda@gmail.com
Teaching Assistant: Office Hours/Office: e-mail:	Steven Zeng M 1 – 3 P.M. stevennz@ucla.edu

Course Outline (subject to changes by the instructor)

Week	Day	Chapters	Subject
1	6/22/2020	1	Introduction/C++ Review
	6/24/2020	3	Pointers, Dynamic Arrays, Resource Management
2	6/29/2020	4, 8	Linked Lists
	7/1/2020	4, 8	Linked Lists (continued)
3	7/6/2020	6, 7, 13, 14	Stacks and Queues
	7/8/2020	3	Advanced C++, Inheritance, Polymorphism
4	7/13/2020	N/A	Midterm #1
	7/15/2020	2, 5	Recursion
5	7/20/2020	2, 5	Recursion (continued)
	7/22/2020	Interludes 6, 7	Templates, Iterators, STL
6	7/27/2020	10, 11	Algorithmic Efficiency, Sorting
	7/29/2020	11, 15	Trees
7	8/3/2020	N/A	Midterm #2
	8/5/2020	18	Hash Tables, Tree-Based Tables
8	8/10/2020	17, 20	Heaps
	8/12/2020	20	Graphs, Modern C++
9	8/17/2020	N/A	Review
	8/19/2020	N/A	Final