

PIC 10A – Introduction to Programming – Spring 2022 Syllabus

Instructor: Weiqi Chu
E-mail: chu@math.ucla.edu
Office: Mathematical Sciences 5352
Class Times: MWF 10-10:50 am
TAs: TBA on Canvas (Bruin Learn)
Office Hours: TBA on Canvas (Bruin Learn)

Textbook:

Big C++: Late Objects, 3rd Edition by Cay S. Horstmann

Textbook is required (a used copy or electronic version also works). Lecture slides will also be provided after each lecture.

Course Content:

(5) Lecture, three hours; discussion, two hours. No prior programming experience assumed. Basic principles of programming, integrated development environments, using C++; algorithmic, procedural problem solving; variables, assignment, fundamental types (int, double, char, bool, size_t, etc.), casting; cin, cout; control flow; the string class, the vector class; functions, function overloading, references; classes, constructor initializer lists; const correctness, source code organization, well-documented code; C-style arrays, pointers (but no memory allocation). P/NP or letter grading.

10A serves as the introductory programming class for the Program In Computing (PIC). C++ is an excellent language for introducing newbies to coding because many of the fundamental concepts (data types and memory management, for example) are addressed explicitly by the language. Once these concepts are well understood, other languages are easier and quicker to learn, and one can better understand the advantages and disadvantages of different languages for solving different problems. 10A covers a lot of ground. Upon completion of the class, a student will be familiar with console input/output, control flow, functions, and classes (object oriented programming). An ambitious and imaginative student will already be able to implement their own interesting applications of coding elegantly.

Course Format:

The intent for instruction is to be fully in person, which includes lectures, discuss sessions, exams, and office hours. Lectures and discussion sessions will not be recorded. All homework assignments are given online and submitted online via Gradescope.

Grading and Examination Policies:

The higher grade from two grading schemes will be used for everyone. Students must complete the final exam to pass. If students cannot take the final due to medical emergencies (including COVID related illness), an incomplete grade will be given until students complete the final exam in the following quarter.

Scheme I		Scheme II	
Homework	50%	Homework	50%
Midterm Exam	13%	Midterm Exam	0%
Final Exam	36%	Final Exam	49%
Participation*	1%	Participation*	1%
Total	100%	Total	100%

Participation*: Students who complete autonomous MyUCLA course evaluation surveys will receive full credits for participation.

The corresponding letter grades are listed here.

Letter grade			
A	[93,100]	C+	[77,80)
A-	[90,93)	C	[73,77)
B+	[87,90)	C-	[70,73)
B	[83,87)	D	[60,70)
B-	[80,83)	F	[0,60)

Homework:

- There will be 8 homework assignments in total. The one lowest homework score will be dropped.
- The first homework assignment will be due at 5pm Friday of Week 3. The homework assignments will be posted to BruinLearn. You will need to log on to view materials.
- All homework assignments must be submitted on Gradescope with no exception. You can always modify your submission before the deadline. Only the latest submission will be graded. Emails about late submission and homework extensions will not warrant a reply.
- Students are encouraged to discuss ideas regarding the homework, but you must write your own code. **You are not allowed to use any existing code no matter where the source is, including code provided by instructor and TAs. All works must be written originally by yourself during this course period.** Never show your code to other students of this class. If similar works are identified, students will be reported to the Dean's office and lose all credits for the cheated homework assignment.
- Visual Studio 2022 is the standard compiler. Homework assignments will be graded according to Visual Studio 2022 alone. If your code does not compile or operate correctly on Visual Studio 2022, marks will be deducted regardless of whether it works on other software.
- Virtual machines installed with Visual Studio 2022 will be accessible to students. You may work on your code using local compilation environments, such as Mac's Xcode, Windows' VS community version or online computing resources (like onlinegdb), but make sure to check your homework with VS 2022 before submission. More information of virtual machines will be posted on Canvas of Week 1.

Exams:

- There will be two in-person closed-book exams for this course. The midterm exam is scheduled from **10am to 10:45am May 6th, 2022**. The final exam is scheduled from **11:30am to 2:30pm on June 6th, 2022**.
- Please be aware of exam dates. There will be no make-ups for missed exams. Remember to check on your schedules to avoid conflict exams. If you miss the midterm exam for any reason, the second grading scheme will be used for your total grades. You must take the final exam to pass.
- The midterm will be returned at the discussion sections. Talk to your TA by the end of the discussion section in which the test is returned. The grade becomes final after you leave the discussion room with your exam. If you miss the discussion section, you can collect your test during the instructor's office hours.
- Instructors reserve the right to contact the students after the exam and ask for additional explanations of solutions for problems. Students who have similar answers will be identified as cheating due to instructors' judgement.
- Deviations from the rules may render the exam void and result in a FAIL for this course.
- You will be asked to sign a written statement to state the exam only shows your own work without help from others. Let me remind you of the Student Conduct Code (it can be found at www.deanofstudents.ucla.edu) See, in particular, Section 102.01 on academic dishonesty.

Participation:

Class attendance is not mandatory, but it is strongly encouraged. Students who complete the autonomous course evaluation on MyUCLA are guaranteed to receive full credits of participation.

PIC Lab:

There is a PIC Lab located in MS 2000. In the PIC Lab, you will find Windows computers equipped with Visual Studio 2022 (VS 2022). This software is particularly important for PIC 10A and 10B because instructors grade code based on how it performs in VS 2022 (not XCode). VS 2022 is not compatible with the Mac OS. Something called “Visual Studio 2022 for Mac” or “Visual Studio Code” is available for Mac OS, but, confusingly, they are either not compatible with C++ language or not a compiler, and you should not use it for this class. The PIC Lab provides a good way for Mac users to check their coding assignments compile as expected on VS 2022.

During the online quarters, a Virtual Lab was created called Apporto. You will all receive an Apporto account in case you need virtual access to VS 2022. The Virtual Lab will be available 24/7 but there will be a limit to the number of simultaneous users, and so you are encouraged to use the physical PIC Lab when possible.

Academic Integrity:

In order to ensure that all students have a fair and equal opportunity to succeed in this course, the Math Department is committed to enforcing the University’s academic integrity policy. In this course, academic misconduct includes, but is not limited to:

- Copying the work/code of another student on an exam or assignment;
- Copying the work/code from any online resources on an exam or assignment;
- Passing off the work of another individual as your own;
- Using non-approved devices or aids on exams;
- Having unauthorized possession of exams;
- Engaging in deception in order to extend or reschedule an exam, or assignment;
- Facilitating acts of academic misconduct by others.

You are encouraged to discuss aspects of the course with other students. You may also discuss the homework assignments in general terms with others. By general terms, I mean discuss ideas and plans of attack for solving the problems, not code. Don’t show your code to another student. **All code and solutions must be originally written by yourself during the course period.** This will allow you to truly understand what you are learning. **Students are not allowed to use any existing code no matter where the source is, including code provided by instructor and TAs.** Homework solutions will be monitored for plagiarism. Cheating of any kind is not tolerated. Please do not endanger your entire academic career by cheating.

Special Needs:

Any students seeking disability services please register with the CAE as soon as possible. If you are already registered and require assistance for the quarter, please release your accommodation letters by the end of week one. Please also be aware that I can only provide special accommodations after I receive the official letter from CAE (<https://www.cae.ucla.edu/>).

COVID-19 related safety guidelines

As the course will be in person, I think it is necessary to remind you UCLA COVID-19 related regulations. Ensuring a safer campus depends on each of us following the latest UCLA health and safety guidelines. While campus policies must be modified to address changing local, state, and national orders and guidance, the most current information is available at covid-19.ucla.edu.

At present, each of us:

- Is responsible, regardless of vaccination status, for wearing an [approved mask](#) that fully covers our nose and mouth for the duration of class, office hours, or other course-related activity.
 - Disposable masks are available at the Wooden Center for anyone unable to obtain a mask or who has forgotten to bring one to campus.
 - Appropriate masks include two-ply woven fabric masks, surgical masks, non-woven KN95 masks, and N95 respirators.
 - Please note that scarves, balaclavas/ski masks, single-layer fabric masks and neck gaiters, bandanas, and turtleneck collars are **not** compliant.

- For those that have a disability-related reason not to wear a mask, you can contact the [Center for Accessible Education](#) (CAE) to obtain an approved accommodation and any appropriate alternative mitigation measures to be sent to instructors.
- Must be fully vaccinated or have submitted an exception request. Per the COVID-19 Response and Recovery Task Force, unvaccinated students with pending or approved exceptions must comply with [twice-weekly testing](#).
- Is required to complete [daily symptom checks prior to coming to campus](#), regardless of vaccination status, and must stay home if you are not cleared by the symptom survey and/or are advised by the Exposure Management Team to quarantine or isolate.
- Will refrain from eating meals in the classroom, except for those with approved accommodations. If you need to eat or drink something for medical reasons, please do so quickly and then put on your mask.

Be advised that refusal to comply with current campus directives related to COVID-19 mitigation may result in dismissal from the classroom and referral to the Office of Student Conduct. If you have any questions or concerns about UCLA's COVID-19 protocol, go to <https://covid-19.ucla.edu/information-for-students/>; if you have any questions specific to this course, I am happy to talk further. Thank you for protecting your fellow Bruins!

Tentative Schedule of Lectures

Approximately 5 lectures on Chapter 2

1. Hello world; cout; variables; assignment
2. Fundamental types (int, double, bool, char); casting using static_cast; limitation of numeric types (overflow, underflow);
3. Casting; round-off error; arithmetic and math library; const keyword
4. Cin/cout; the string class
 - indexing with operator[] and at
 - length, substr, push_back, pop_back
 - find, rfind, concatenation (+)
5. Getline; input buffer; formatted output
 - cin.ignore(), cin.get(), cin.unget(), cin.peek(), cin.fail()
 - setw, fixed, set_precision

Approximately 6 lectures on Chapter 3 & 4

1. Decisions; logical operators;
 - If, if-else, else if
 - <, >, ==
2. Nested branches; logical operator (and, or, not); input validation;
3. Loops (while, for, do-while); hand-tracing
4. More with loops
 - Nested loops
 - break, continue
5. Random number generator and loop algorithms
 - rand(), srand()
 - Processing inputs
 - Find min/max value, compute sum/average, count matches, etc.

Approximately 3 lectures on Chapter 5

1. Functions
 - function signatures, function definitions, function comments
 - void, return keywords

- parameters and arguments
- scopes, forward declaration
- 2. References and const correctness
 - when to pass by value
 - when to pass by reference
 - when to pass by reference to const

Approximately 3 lectures on Chapter 6

1. `std::vector` class
 - `at`, `operator[]`, `empty`, `size`, `clear`, `push_back`, `pop_back`
2. Arrays
3. Two-dimensional vectors and arrays

Approximately 3 lectures on Chapter 9 -- classes

1. Classes
 - Class interfaces, public, private categories
 - member variables and member functions
 - constructors and constructor initializer lists
 - mutators and accessors
 - const correctness for member functions
2. Default values
 - for functions
 - for member functions
 - source code organization, separate compilation

Approximately 3 lectures on Chapter 7 (structures not covered)

1. Pointers, memory allocation (`new`, `delete`), pointer arithmetic
2. Dynamic arrays, array pointer duality law
3. Memory leak, dangling pointers.

Approximately 2 lectures on Chapter 8 streams

1. File streams
 - `std::ifstream`, `std::ofstream`, random access,
2. String streams
 - `std::istringstream`, `std::ostringstream`, conversion between numbers and strings