

COMPSCI 198C:
Introduction to the C Programming Language
1 Credit





Contact Us 	Meet with Us 	Class Location, Day, and Time 	Course Materials 
Tim Richards richards@cs.umass.edu	Ask question on Piazza	Online	Programming in C (3rd edition)
Joe Chiu joechiu@cs.umass.edu			See web pages for posted slides, and Canvas for video recordings.
TAs/UCAs	See Canvas for the TA/UCA office hour schedule. Check Piazza for any updates.	Online.	See Canvas

Table of Contents

Course Description	2
Prerequisites	2
Who are we?	2
How to succeed in this course	3
Equality Statement	3
Accommodations	3
Office Hours	3
Attendance and Participation	3
Communication	3
Rules for Success (Student Responsibilities)	4
Lateness Policy	4
Course Staff Responsibilities	4
Virtual Environment	4
Grading and Course Assignments	5
Final Grades	5
Course Incompletes	5

Course Description

Official description: This practicum assumes general background and experience in computer programming (such as that provided by COMPSCI 121 or a similar introductory programming course) and some knowledge of data structures. Content will include basic C data types, declarations, expressions, statements, and functions; simple use of macros; some common library calls (such as formatted input/output); basic pointer manipulation using linked lists; and introduction to using standard tools (gcc and make).

Prerequisites

No Prerequisites.

Who are we?



- My name is Tim Richards. I have been part of the teaching faculty in CICS for the past 10 years as well as the Undergraduate Program Director.
- I love Computer Science, teaching, raising chickens, and traveling. My two top locations to visit are Italy and Portugal. I am also a big fan of most European countries.
- I have a BA in CS from Clark University, MS in CS, and PhD in CS from UMass.



- My name is Joe Chiu.
- I'm originally from Taiwan, so I can speak mandarin.
- I like to play board games and Pokemon Go.
- I have a BA in CS, MS in CS, and I got my PhD in UMass CS.

How to succeed in this course

Your success in this class is important to us. We all learn differently and bring different strengths and needs to the class. If there are aspects of the course that prevent you from learning or make you feel excluded, please let us know as soon as possible. Together we'll develop strategies to meet both your needs and the requirements of the course. There are also a range of resources on campus, including:

- Learning Resource Center - <http://www.umass.edu/lrc>
- Center for Counseling and Psychological Health (CCPH) - <http://www.umass.edu/counseling>
- English as a Second Language (ESL) Program - <http://www.umass.edu/esl>

Statement of Inclusivity

The instructors are dedicated to establishing a learning environment that promotes diversity of the students including (but not limited to) race, class, culture, religion, gender, sexual identity, and

physical ability. We respect everyone's right to be addressed by the name and pronouns that they choose. You can indicate your preferred/chosen first name and pronouns on SPIRE, which appear on class rosters. A student's chosen name and pronouns are to be respected at all times in the classroom. It is important that this is a safe classroom environment. We will practice being generous and respectful members of our class and computer science community. Please let us know immediately if you notice discriminatory behavior in this class, or feel discriminated against.

Accommodations

As this is mostly a self-paced course its timeline is pre-built with accommodations already included. In order to ensure that all accommodations are addressed for everyone, the amount of time you have for programming assignments is at least 3x the normal amount of time we would expect it would take for a student to complete and you have an unlimited amount of time and retakes to complete quizzes. Most accommodations for extra time on assignments/exams/quizzes are either 1.5x or 2x. We decided on at least 3x and you also have more time if you choose to start early when you complete a previous assignment. Although you are welcome to register with disability services for this course, your accommodations have already been satisfied by design.

If you have further questions or concerns, please do not hesitate to ask.

Office Hours (Fall and Spring Semesters only)



Office hours are an important part in supporting you throughout this course. Even if you don't have specific questions, needs, and concerns, we would love to meet up with you at least once during this semester. There are a couple of options to meet up: stop by our Zoom office during the office hours, or, if these times don't work for you, please email us and we will find another day and/or time. In some semesters office hours may be held by UCAs.

“Non-office” Hours (Winter and Summer offerings)

Outside of the regular semesters, we will not be hosting specific office hours. However, we invite you to post on Piazza to request a help session as needed. Between us, someone should be available, just not on a specific time table!

Attendance and Participation

This course is entirely asynchronous and self-paced.

Communication

We will be using the Piazza discussion forum system for communication. The discussion forum should be your first choice for asking questions as others most certainly have the same question. You should check the discussion forum before asking your question to see if the same question has already been posted. We will tend not to answer questions directed towards us that have already been answered in the discussion forum. Think before you post. We expect you to do a reasonable amount of thinking to try to solve your problems before posting for help. Make sure you understand the rules and try to be articulate and clear with your post (again, think before you post). You should post questions related to assignments early rather than waiting until the last minute. If

you post a question too close to an assignment deadline, you may not receive an answer before that deadline. Course staff are expected to answer questions Monday through Friday. Do not expect prompt answers on Saturday, Sunday, and scheduled holidays and breaks.

Rules for Success (Student Responsibilities)

This course has six important rules. If you choose to follow these rules, your odds of learning the material and achieving a good grade in this course will improve greatly. You should schedule approximately 3~4 hours per week to work for this course.

1. Do the assigned reading from the book, preferably *before* class.
2. Read the assignment documentation early.
3. Do your work on time, submit your work on time, and make sure you submitted the correct work (for more details about deadlines please review the [Lateness Policy](#) section).
4. Communicate with other students in the course, the instructor, and any other course staff for help (but see the honesty policy about what constitutes legitimate help!)
5. Be honest in the work you do and the submissions you make.
6. Communicate with us and others in the course with respect and understanding.

Lateness Policy

Deadlines for the challenges are listed on Gradescope. We accept work after the deadline (up to a cutoff that you can also find on Gradescope - it's at the end of the semester), but assess a penalty of 10%, plus 5% for each full week late, up to a total of 25%. This is in terms of the total points available, not a percentage of what you achieved. Given that the minimum grade requirement is 50%, it remains possible to pass each challenge even if it is maximally late - but students who leave too much work to the end of the semester historically show a low chance of successful completion.

Course Staff Responsibilities

You can expect from us:

1. Timely release of course assignments.
2. Timely release of scores achieved on course assignments.
3. Response to questions in the discussion forums in a reasonable amount of time.
4. Respectfulness of your ideas and valuing of the diversity you bring to the class.
5. Openness to dialogue that challenges us.
6. Presence during our stated office hours.
7. Proper running of the course.

Programming Environment

All the challenges in this class have to be implemented in the 32-bit x86/Linux environment we provide in [edlab](#). It is important that you have the precise environment so you can follow along in lectures and complete the challenges correctly.

If you use a command prompt, type `ssh <NETID>@elinux.cs.umass.edu`. After entering your password, your command prompt should look like this:

```
elinux1 ~) >
```

Grading and Course Assignments

Challenges (programming assignments)	90%
Quizzes	10%
Total:	100%

This course has 12 challenges: 1 through 8, and 10 through 13. Challenge numbers match module numbers, except that Module 12 has two challenges, 12 and 13. Each challenge is a programming assignment designed around the topics we are studying. All challenges must be implemented in the C programming language. ***This scheme may be adjusted.***

Grades for challenges are updated to Canvas periodically (a manual, not automatic process), typically weekly and then more frequently toward the end of the course. A pinned Piazza post indicates when the grades were most recently updated.

Final Grades

This course offers Pass/Fail grading only. To evaluate your understanding of the course content we will use scores achieved on each of the above assessment components. Your final grade will convey *what you demonstrated that you know* from the course and *how well you demonstrate that knowledge*. Missing and late assignments can have a dramatic impact on your final grade so it is important that you are attentive to submission deadlines and avoid any missing work. ***This grading scheme may be adjusted based on the overall performance of students in the course.***

The requirement to pass is: at least 70% average, at least 50% on each quiz and challenge and at least 50% on ANY 5 challenges before the Mid-Course Deadline. (The 50% is after any late penalties.) The deadline date can be found on Canvas under the Requirements and Grading section.

The late policy on challenges is 10% off for being late, plus an additional 5% for each full 7 days later, up to a maximum of 25%. x% off means losing x% of the possible points.

In this calculation, we drop one lowest challenge score before applying the rules above.

Course Incompletes

Students who are unable to complete course requirements within the allotted time because of severe medical or personal problems may request a grade of Incomplete from the instructor of the course. Incomplete grades are warranted only if a student is passing the course at the time of the request and if the course requirements can be completed by the end of the following semester. Furthermore, an incomplete will be granted if at least 75% of the work has been completed for the course. Otherwise, the recommended course of action is to *withdraw* and retake the course in the

future. Please see the [Academic Regulations](#) Section IV Grading System and Credit Guidelines for further details.

Note: An incomplete means you are on your own to complete the material agreed upon by the instructor of this course. Do not expect additional help or one-on-one teaching of the material past the course completion date. It is your responsibility to complete the remaining material.

Academic Honesty

It is very important in all courses that you are honest in all the work that you complete. You **may** discuss assignments with other students - in fact we encourage this as a learning experience. But what you turn in must be your work. Copying is not allowed, and collaboration so close that it **looks like** copying is not allowed. Remember to **tell us who you worked with** as well. Also, the use of ChatGPT and similar AI text generators is prohibited according to the [Academic Honesty Policy](#). We do not allow the use of ChatGPT and similar AI text generators because it bases the code written by other people. We do employ tools such as similarity checkers, and be aware, they are not easily fooled!

If you copy on challenges or exams you are doing a disservice to yourself, the instructors for the course, the College of Information and Computer Sciences, the University of Massachusetts, and your future. We design our courses to provide you the necessary understanding and skill that will make you an excellent computer scientist. Assignments are designed to apply and test your knowledge and understanding of the material. Plagiarism and academic honesty of any sort may seem like an easy way to solve an immediate problem (which it is not), however, it can have a substantial negative impact on your career as a computer science student. There are many computing jobs out there and many more people working hard to get those positions. If you do not know your stuff you will have a very difficult time finding - and keeping - a job. Please take this seriously.

We will carefully review your submissions automatically and manually to verify that “cheating” has not taken place. If you are suspected of plagiarism, we will follow an informal path to determine if academic dishonesty has taken place, and you may receive an F for the course and have a mark on your permanent record at UMass. This will disrupt your schedule for completing courses and may lead to your not completing your degree in a timely fashion. Please review carefully the [Academic Honesty Policy](#), [Avoiding Plagiarism](#), and the [Academic Honesty Flowchart](#) to understand what academic honesty is, how you can avoid it, and the procedure we will follow if you are under suspicion. If you have questions or are unsure if something constitutes plagiarism, please reach out to us. A good rule of thumb to avoid plagiarism is: don't leave a meeting with other students/helpers (except course instructors) with anything written down (in whatever form) that you did not have when you entered the meeting. That way the learning is in your brain.

Every assignment implicitly includes a “contract” that you sign virtually by submitting your assignment. By “signing” this contract you indicate that you have read all the documents on this website and any links to academic honesty associated with the university. Your signature indicates that you completely understand the policies in place and that you have not plagiarized.

Final note: it is a violation of policy of the College of Information and Computer Sciences, and of this course, to post answers, code, etc., in public places that might facilitate someone else's

copying them. Such violations are treated under the Academic Honesty Policy.

Title IX

If you have been the victim of sexual violence, gender discrimination, or sexual harassment, the university can provide you with a variety of support resources and accommodations. UMass is committed to providing these resources with minimal impact and costs to survivors on a case-by-case basis. Resources are available to survivors with or without them filing a complaint. No upfront costs are charged to any currently enrolled students for University Health Services or the Center for Counseling and Psychological Health, and no fees exist for services in the Dean of Students Office, the Center for Women and Community, Student Legal Services, or by live-in residential staff.

Course Technology

In this course, we use **Canvas**, **Piazza**, and **Gradescope** systems to enrich your learning experiences.

Canvas

Canvas is a Learning Management System (LMS) used for posting course syllabus, materials, assignments, and grades. You are enrolled in the Canvas course through Spire. You should have had an account automatically created for you at the beginning of the course. If you are unable to log in to Canvas, you should contact the professors.

The Canvas webpage, when expanded, shows the weekly topics and material covered in lectures. Grades will be available through the Canvas gradebook. You should orient yourself with how to find it and stay on top of your grades to resolve any grading issues in a timely manner.

Piazza

We'll use Piazza as the main hub for communication. You can ask questions in Piazza about the material in this course or for administrative matters. All content related questions should be posted on Piazza and not asked via email to the course staff. If you have strong feelings or opinions, positive or negative, about any aspect of this course, the professors encourage you to let them know in a private post which can be followed up if necessary.

Gradescope

We use Gradescope for grading your challenges. Gradescope allows us to provide fast and accurate feedback on your work. Challenges are graded automatically using Gradescope autograder. Before the deadline you can submit as many times as you need, so submit early and often to ensure you have something in before the deadline. Become familiar with Gradescope and verify that your

submission has been properly uploaded before the deadline. Use OneDrive, DropBox, Google Drive, or some other backup software to ensure that your work is not lost in the event of a computer failure

The Gradescope autograder will provide you with some limited feedback on your submissions: does it compile, does it pass the public and private tests, what your score is, etc. The autograder does not provide detailed feedback.

Course Topics

(Check Gradescope for due dates, the due date below is based on regular semester)

Module 01: Course Introduction

Module 1, Lesson 1, Course Introduction

Module 1, Lesson 2, History of C

Module 1, Lesson 3, Linux Command Line Basics with SSH and SFTP

Module 1 (challenge: starter_project) - due on week 2

Module 02: C Basics

Module 2, Lesson 1, C Basics: printf and scanf

Module 2, Lesson 2, C Basics: branches and loops

Module 2, Lesson 3, GDB Debugging

Module 2 (challenge: unit_conversion) - due on week 3

Module 03: Arrays and Functions

Module 3, Lesson 1, Arrays

Module 3, Lesson 2, Functions

Module 3, Lesson 3, Strings

Module 3 (challenge: playing_with_arrays) - due on week 4

Module 04: Structs

Module 4, Lesson 1, Structs - part 1

Module 4, Lesson 2, Structs - part 2

Module 4, Lesson 3, Enum

Module 4 (challenge: event_structure) - due on week 5

Module 05: Number Representation

Module 5, Lesson 1, Numbers: Binary, Hexadecimal, Conversions

Module 5, Lesson 2, Numbers: Computing in Binary, Negative Numbers

Module 5, Lesson 3, Representation: C Integer Types and Casts

Module 5, Lesson 4, Representation: Bit Manipulation

Module 5 (challenge: bits & bytes) - due on week 6

Module 06: Pointers

Module 6, Lesson 1, Basic Pointers

Module 6, Lesson 2, Structure Pointers

Module 6 (challenge: military_time) - due on week 7

Module 07: More Pointers

Module 7, Lesson 1, Pointer Arithmetic

Module 7, Lesson 2, Static Linked Lists

Module 7 (challenge: Binary Search Tree) - due on week 8

Module 08: Dynamic Allocation

Module 8, Lesson 1, Malloc & Calloc

Module 8, Lesson 2, Dynamic Linked Lists

Module 8 (challenge: credential_list) - due on week 9

Module 09: Preprocessor

Module 9, Lesson 1, The Preprocessor

Module 9, Lesson 2, #define

Module 9, Lesson 3, #include

Module 10: String-oriented File Input/Output

Module 10, Lesson 1, Intro and printf deeper dive

Module 10, Lesson 2, scanf deeper dive

Module 10, Lesson 3, More file functions

Module 10 (challenge: math_parsing) - due on week 11

Module 11: Program Inputs

Module 11, Lesson 1, Program Inputs: argc, argv, and the environment

Module 11, Lesson 2, Program Inputs: getopt

Module 11, Lesson 3, More String Functions

Module 11 (challenge: args and strings) - due on week 12

Module 12: Larger Programs

Module 12, Lesson 1, Larger Programs

Module 12, Lesson 2, The make utility

Module 12 (challenge: compare_gpa) - due on week 13

Module 12,(challenge: make) - due on week 13