

## 1 Prerequisites and description

Prerequisite: C or better in CMSC 132  
Corequisite: CMSC 250 (or equivalent)  
Credits: 4 credits

This course introduces many of the concepts that lie behind software, such as hardware, memory layout, memory management, and operating systems. It explains how these concepts affect the design of software systems. This course provides a transition from the Java environment of the preceding two courses to programming in C.

The objective of the course is to develop a good working knowledge of how to program in the C programming language, to understand how to write programs with explicit memory allocation and deallocation, and to see an introduction of how to write systems programs.

## 2 Instructor contact information

Name: Larry Herman  
Office: 1111 A. V. Williams  
Phone: (301) 405-2762

Office hours will be provided in a separate handout shortly.

## 3 Teaching assistants and TA office hour information

There is still, as of this writing, some uncertainty about the number of TAs to be assigned to the course, which should be resolved in a day or two. Therefore information about the TAs, including their office hours schedule, is still not definite, and will be provided in a separate handout shortly. The TA teaching your discussion section may still change.

While the TAs will provide assistance with assignments during office hours, you are ultimately responsible for developing and debugging your own programs, which is your coursework that you're receiving a grade for. You should therefore not rely on the instructional staff to make your projects work.

## 4 Textbooks

**Required text:** *Pointers on C*, Reek, Addison-Wesley 1998; ISBN 0-673-99986-6.

**Recommended text:** *Computer Systems: A Programmer's Perspective*, Bryant and O'Hallaron, Prentice Hall, 2003; ISBN 0-13-034074-X.

## 5 Class webpage, computing environment, and submission and grades systems

Various course materials will be made available on the class webpage, which is at the following URL:

[www.cs.umd.edu/class/spring2006/cmssc212](http://www.cs.umd.edu/class/spring2006/cmssc212)

Accessing the webpage will require an ID and password to be provided in class. All students are expected to check the webpage frequently (at least every other day), because important corrections or clarifications to projects may be made there.

Programming will be done on the OIT Grace Cluster, in particular on the two Linux machines which can be accessed at [linux.grace.umd.edu](http://linux.grace.umd.edu). Students will use their own Glue accounts to access the Grace cluster and do coursework, so students who don't have a Glue account should request one online immediately at [www.oit.umd.edu/new](http://www.oit.umd.edu/new).

Project submission will be done using the same CMSC project submission and testing server used in the preceding courses, located at <https://submit.cs.umd.edu>, but using a different mechanism to turn projects in to the server than what was used in the preceding courses. Details will be provided as part of the first project assignment. Unlike the preceding courses, the program development environment will not be the Eclipse IDE, but rather command-line use of Linux. Grades will be recorded and available on the CMSC grades server at <https://grades.cs.umd.edu>.

## 6 Exam and final dates

Midterm exams will be held during the lecture time, but possibly in a different location, to be announced in class. The final exam date appearing in the University's Schedule of Classes and on Testudo is unfortunately **incorrect** however, and will be rescheduled by the Registrations office. However, the Registrations office says they will not be able to set the actual final date until halfway through the semester. The final (when its date and time are actually known) will be rescheduled **only** for students having another final at **exactly** the same time, or for students with more than three final exams scheduled on the same day. If either of these situations applies to you, you must inform the instructor **within two weeks** of the final exam time being announced in class for any allowances to be made.

The following dates are when the midterm exams are expected to fall, so at this time you should plan on having exams on these dates. Note however that if it becomes necessary these dates could be adjusted depending upon lecture progress during the semester or other factors, therefore these dates will either be confirmed or adjusted as necessary, to be announced in class.

Exam #1: Tuesday, March 7 (during the lecture time)

Exam #2: Tuesday, April 18 (during the lecture time)

Final exam: currently unknown; the time given in the Schedule of Classes is **incorrect**

## 7 Attendance and general grading policies

Students are responsible for all material covered, and all announcements, deadlines, policies, etc., discussed in lecture and discussion section, regardless of whether they were in class or not. **It's understood that students may occasionally have to miss class for various reasons, but email and office hours are not intended as a replacement for class attendance. Consequently, only students who typically and regularly attend class will receive assistance during office hours.**

Coursework will count toward the final grade according to the following percentages:

Midterms:	two midterms	30%	(equally weighted; 15% each)
Final:	will be comprehensive	25%	
Programming projects:	six expected coding assignments	35%	
Quizzes	in discussion section	10%	(equally weighted)

All projects will be graded out of 100 points, but depending upon their relative difficulty, which can't easily be estimated in advance, they may not be weighted equally. Therefore the relative weights of the projects will be given toward the end of the semester.

Quizzes will be given in discussion section and will cover discussion and lecture material.

Any request for reconsideration of the grading on any coursework **must** be submitted **within one week** of when it is returned. Exam regrading requests must be made in writing. Any coursework submitted for reconsideration may be regraded in its entirety, which could result in a lower score if warranted.

Final course grades will be curved as necessary, based on each student's total numeric score for all coursework at the end of the semester.

## 8 Project submission and grading policies

### 8.1 Project submission method and deadlines

Projects will be submitted electronically using the CMSC project submission and testing server. **No attempt** to submit a project using any other means (such as an emailed project) be considered. Only the projects electronically submitted via the submit server can be graded; it is **each student's responsibility** to test their program and **verify that it works properly** before submitting.

All projects will be due at 10:00:00 p.m. on the day indicated on the project assignment. Projects may be submitted up to three days late, with a 15-point late penalty deducted per each day (24-hour period) that the project is late. Submission deadlines are **firm**, and other than very limited situations such as those described in Section 9.1 exceptions cannot be made. Note there is **no grace period** for project submissions—deadlines will be enforced at exactly 10:00:00 p.m. the day a project is due, and every 24 hours later for the next three days. The project submission server will still accept projects more than three days late and may report numeric results for them, but we will not give them any credit in our recorded grades.

Project extensions will not be given to individual students as a result of system problems, network problems, power outages, etc., so **do not leave** submitting a project until the night it is due. It is strongly suggested you finish and

submit your program **at least** one day early, to allow time to reread the project assignment, to insure you have not missed anything which could cause you to lose credit on the project.

## 8.2 Project grading policies

During each project assignment certain public tests will be made available on the project submission server. Projects will also be run on various other test cases which will not be provided in advance; these will test conditions the public tests do not. Projects will be graded out of 100 points as follows:

- A project which was not submitted will receive a score of zero.
- The 15 point late penalty will be applied for each day late the project was submitted, up to a maximum 45 point deduction.
- The project's score will then be computed as follows:

results of public and non-public (secret) tests	85 points
style and documentation	15 points (prorated)

All credit for any test cases will be lost if a project does not satisfy that test case; i.e., partial credit will not be given for individual test cases.

- It is expected that the public tests will typically count for around 50 points of a project assignment, and the remaining secret tests for 35 points (with style and documentation always counting for 15 points as mentioned).
- If you make more than one submission for any project (on-time, one day late, two days late, or three days late), or make more than one submission on any day, the project submission server will record the results of all of the submission and the highest of their scores is what you will receive.

Note: this is a programming course which is a required part of the curriculum for CMSC majors and minors and for ENCE majors, and which teaches languages and concepts essential for later CMSC courses. As a result, the ability to submit reasonably-successful versions of the projects is necessary. The instructor reserves the right to fail, regardless of overall numeric score, students who have not shown a good-faith effort to complete all programming assignments. A good-faith effort to complete all programming assignments would typically be considered to mean that a student's projects have correct results for almost all of the public tests for all project assignments. If in doubt, discuss your situation with the instructor in person.

Any hardcoding in a project assignment will result in a score of zero for that project (and consequently the student would be in jeopardy of not passing due to the preceding paragraph). Hardcoding refers to attempting to make a program appear as if it works correctly and actually calculates and computes correct results, when for some reason it actually does not do so. Examples would include a program which prints the desired output instead of computing it, or a program which works only because it takes advantage of properties which the public test cases happen to have, etc. These are only a few examples; if you have any question about whether a particular situation would constitute hardcoding be sure to ask ahead of time.

## 8.3 Issues regarding computing resources

Projects can be developed on the Linux hosts on the OIT Grace UNIX Cluster. You may use any other available system, but all project submissions **must** work correctly using the gcc C compiler the Grace cluster, using the compilation method and compiler options which may be specified from project to project. Because different C compilers or different versions of the gcc compiler may be installed elsewhere, a program may work perfectly on one system, yet not work at all on the Linux Grace machines. The program you submit will be graded based on its results on the Linux Grace machines, so having a working version on another system at any other time (or even another working version in your Glue account) can not be considered. No consideration in grading can be made for errors made in transferring files, or submitting the wrong version of your project. If you want to write any project on another system you are strongly recommended to complete it **several days early**, to have time to address any problems arising.

# 9 Excused absences and accommodations

## 9.1 Excused absences

Besides the policies in this syllabus, various University policies may apply to students during the semester. Various policies which may be relevant appear in the Undergraduate Catalog at [www.umd.edu/catalog](http://www.umd.edu/catalog).

If you experience difficulty during the semester keeping up with the academic demands of your courses, you may consider contacting the Learning Assistance Service in 2201 Shoemaker Building at (301) 314-7693. Their educational counselors can help with time management issues, reading, note-taking, and exam preparation skills.

Missing a quiz or exam an exam for reasons such as illness, religious observance, participation in required university activities, or family or personal emergency (such as a serious automobile accident or close relative’s funeral) will be considered to be an excused absence. However, students requesting an excused absence for any reason must apply in writing and must furnish documentary support for the assertion that the absence qualifies as an excused absence. For an absence due to medical reasons, for example, documentation would be from a health care professional who treated the student. In cases of illness simply being seen by a health professional is insufficient– medical documentation must state that a student was **incapacitated** and therefore unable to attend for an excused absence to be justified. For medical absences the documentation **must** include the phone number of the health care professional and **must** explicitly indicate the exact dates or times of incapacitation due to illness. The dates of incapacitation must **include** the date of the missed exam preceding an exam or quiz may not be considered to justify an excused absence. Self–documentation of illness is not sufficient support to excuse an absence. Excused absences will not be given unless documentation as described is provided. **If you become ill, keep in mind that the University Health Center will not provide medical documentation.**

It is the University’s policy to provide accommodations for students with religious observances conflicting with exams, but it is the **student’s responsibility** to inform the instructor **in advance** of intended religious observances. Written notice must be provided **immediately** upon an exam date being announced or confirmed in order for an absence to be excused, and if a known conflict exists with one of the planned midterm dates appearing below, notice **must** be given prior to the end of the schedule adjustment period. Excused absences for quizzes will also be considered in the case of religious obligation; contact the instructor as soon as possible after the quiz date.

There will be no makeups for missed quizzes– with an excused absence the score for a missed quiz will be counted by averaging the student’s scores for the other quizzes. When a student has an excused absence for an exam the score will be determined either by averaging their scores for the other exams (possibly a weighted average), or by giving a makeup exam. However, unless **immediate** notice is given as early as possible of the reason for any missed coursework, an excused absence may not be granted.

A student who might miss an exam for any reason other than those mentioned above must contact the instructor **in advance** to discuss the circumstances. An instructor is not under obligation to offer a substitute assignment or to give a student a makeup assessment unless the failure to perform was due to an excused absence.

The policies for excused absences do not apply to project assignments. Projects will be assigned with sufficient time to be completed by students who have a reasonable understanding of the necessary material and begin promptly. In cases of **extremely serious documented illness of lengthy duration** or other **protracted, severe emergency situations**, the instructor may consider extensions on project assignments, depending upon the specific circumstances.

## 9.2 Students with disabilities

Students with disabilities who have been certified by Disability Support Services as needing any type of special accommodations should see the instructor as soon as possible, during the schedule adjustment period.

All arrangements for exam accommodations as a result of disability **must** be made and arranged with the instructor **at least** three business days prior to the exam date, or accommodations **will not** be made.

## 10 Academic integrity statement

The Campus Senate has adopted a policy asking students to include the following statement on each examination or assignment in every course: “I pledge on my honor that I have not given or received any unauthorized assistance on this examination (or assignment).” Consequently, you will be requested to include this pledge on each exam and project.

Please carefully read the Office of Information Technology’s policy regarding acceptable use of computer accounts provided for instructional use at [www.nethics.umd.edu/aug](http://www.nethics.umd.edu/aug).

Note that programming projects are to be written **INDIVIDUALLY**, therefore cooperation or use of unauthorized materials on projects is a violation of the University’s Code of Academic Integrity. **Any evidence** of this, or of unacceptable use of computer accounts, use of unauthorized materials or cooperation on exams or quizzes, or other possible violations of the Honor Code, **will be submitted** to the Student Honor Council, which could result in an XF for the course, suspension, or expulsion.

- For learning the course concepts (including the C programming language), students are welcome to study together or to receive help from anyone else. Students may discuss with others the project requirements, the C language, what was discussed in lecture and discussion section, and general syntax errors. Examples of allowable questions are “What happens in C if you include the same header file in a source file more than once?” or “What does ‘Segmentation fault’ mean?”, because they convey no information about the contents of a student’s project solution.

- When it comes to actually writing a project assignment, other than help from the instructional staff a project must solely and entirely be a student's own work. Working with another student or individual, or using anyone else's work IN ANY WAY except as noted in this paragraph, is a violation of the code of academic integrity and WILL BE REPORTED to the Honor Council. Students may not discuss design of any part of a project with **anyone**, except the instructor or teaching assistants. Examples of questions which students may **not** ask others might be "How did you implement this part of the project?" or "Please look at my code and help me find my stupid syntax error!". Students may not use any disallowed source of information in creating either their project design or code. When writing projects students are free to use ideas or **short** fragments of code from **published** textbooks (for example the texts for this course) or **publicly available** information, if the specific source is cited in a comment in the relevant section of the program.

VIOLATIONS OF THE CODE OF ACADEMIC INTEGRITY MAY INCLUDE, BUT ARE NOT LIMITED TO:

1. Failing to do all or any of the work on a project by yourself, other than assistance from the instructional staff.
2. Using any ideas or any part of another person's project, or copying any other individual's work in any way.
3. Giving any parts or ideas from your project, including test data, to another student.
4. Allowing any other students access to your program on any computer system.
5. Transferring any part of a project to or from another student or individual by any means, electronic or otherwise.

If you have any question about a particular situation or source then consult with the instructor in advance. Should you have difficulty with a programming assignment you should **see the teaching assistants in office hours**, NOT solicit help from anyone else in violation of these rules.

IT IS THE RESPONSIBILITY, UNDER THE HONOR POLICY, OF ANYONE WHO SUSPECTS AN INCIDENT OF ACADEMIC DISHONESTY HAS OCCURRED TO REPORT IT TO THEIR INSTRUCTOR, OR DIRECTLY TO THE HONOR COUNCIL.

Every semester the department has discovered a number of students attempting to cheat on project assignments, in violation of academic integrity requirements. Students' academic careers have been significantly affected by a decision to cheat. Think about whether you want to join them before contemplating cheating, or before helping a friend to cheat.

Students are welcome and encouraged to study and compare or discuss their implementations of the programming projects with any others after they are graded, **provided that** all of the students in question have received nonzero scores for that project assignment, and if that project will not be extended upon in a later project assignment.

## 11 Course topics (SUBJECT TO CHANGE)

The following list of lecture topics may vary according to the pace of lecture, so all times are approximate.

- Introduction: moving from Java to C (2 1/2 weeks)
- Pointers and memory management (3 1/2 weeks)
- I/O (2 weeks)
- Libraries and linking (1 week)
- Program measurement (1 week)
- Primitive data representation (1 week)
- Array representation (2 weeks)
- Implementation of object-oriented concepts (1/2 week)
- Multi-language support (1/2 week)

## 12 Right to change information

Although every effort has been made to be complete and accurate, unforeseen circumstances arising during the semester could require the adjustment of any material given here. Consequently, given due notice to students, the instructor reserves the right to change any information on this syllabus or in other course materials.

## 13 Copyright

All course materials are copyright Larry Herman © 2005. All rights reserved. Students are permitted to use course materials for their own personal use only. Course materials may not be distributed publicly or provided to others (excepting other students in the course), in any way or format.