

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 systems, 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, to understand the UNIX and C memory model of a program as well as what happens when a program is running, and to introduce how to write systems programs.

2 Contact information

2.1 Email contact

Due to time constraints and other factors it is impractical or impossible to provide information or help regarding the programming projects via email, and attempting to do so typically results in incomplete or inadequate information. Therefore please ask questions about projects in person, either during office hours, or before or after class.

In general we will be unable to provide explanation about the course material via email; this is more appropriate for class discussion or personal communication. Please discuss regular course business with us in person, and use email only for issues of an urgent or emergency nature.

2.2 Instructor

Name: Larry Herman
Office: 1111 A. V. Williams
Phone: (301) 405-2762
Email: larry@cs.umd.edu (*)

(*) See Section 2.1 above regarding email.
Office hours will be provided in a separate handout shortly.

2.3 Teaching assistants

<table>
<thead>
<tr>
<th>Name</th>
<th>Duties</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abdel-Hameed (Hameed) Badawy</td>
<td>teaching, 0101 &amp; 0102</td>
<td><a href="mailto:absalam@cs.umd.edu">absalam@cs.umd.edu</a> (*)</td>
</tr>
<tr>
<td>Sam Huang</td>
<td>teaching, 0301 &amp; 0302</td>
<td><a href="mailto:srhuang@cs.umd.edu">srhuang@cs.umd.edu</a> (*)</td>
</tr>
<tr>
<td>Hazem El-Alfy</td>
<td>teaching, 0401 &amp; 0402</td>
<td><a href="mailto:helalfy@cs.umd.edu">helalfy@cs.umd.edu</a> (*)</td>
</tr>
<tr>
<td>Rachel Schwartz</td>
<td>grading</td>
<td><a href="mailto:rachell13@cs.umd.edu">rachell13@cs.umd.edu</a> (*)</td>
</tr>
<tr>
<td>Joao Soares</td>
<td>grading</td>
<td><a href="mailto:joao@cs.umd.edu">joao@cs.umd.edu</a> (*)</td>
</tr>
<tr>
<td>Hailey Lin</td>
<td>undergraduate, office hours</td>
<td><a href="mailto:halin@cs.umd.edu">halin@cs.umd.edu</a> (*)</td>
</tr>
</tbody>
</table>

(*) See Section 2.1 above regarding email.
Office hours will be provided in a separate handout shortly.

While the TAs will provide assistance with assignments during office hours, you are ultimately responsible for developing and debugging your own program, 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.

3 Course topics (subject to change)

The following list of lecture topics may vary according to the pace of lecture, so all times are approximate.
1. Introduction: moving from Java to C (3 weeks)
2. Pointers and memory management (1 week)
3. Dynamic data structures in C (3 weeks)
4. I/O, standard libraries (1 1/2 weeks)
5. Testing (1/2 week)
6. Data representation (1 week)
7. Libraries and linking (1 week)
8. Process control (1 week)
9. Program measurement and optimization (1 week)
10. Data representation and implementation of functions (2 weeks)
11. Sockets (1 week)

4 Textbooks


Students are encouraged to purchase the recommended text, as it’s going to be the required text for CMSC 311 which many students in this course must take later. However, for those who don’t buy it, two copies have been placed on course reserve in the MATH library and can be read there.

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

Various course materials will be made available on the class webpage, which can be accessed via the following link: www.cs.umd.edu/class/spring2009/cmsc212

Accessing the webpage will require an ID and password to be provided in class. Students are expected to check the webpage frequently, especially near the time that projects are due, 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 that can be accessed at 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 yet should request one online immediately at the following link: www.oit.umd.edu/new

Project submission will be done using the same CMSC project submission and testing server used in the preceding courses, which can be reached via the following link: https://submit.cs.umd.edu

However, you will use a different mechanism to turn projects in to the server than was used in the preceding courses; further details will be provided in discussion section. 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 may be checked for correctness on the CMSC lower-level course grade server, which may be reached at: https://grades.cs.umd.edu

6 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 to hear the information 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 be able to receive assistance during office hours.**

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

Midterms:  two midterms  30%  (15% each)
Final: will be comprehensive  25%
Programming projects: six expected coding assignments  30%
Quizzes: in discussion section  10% (equally weighted)
Discussion section exercises: to be done during discussion section  5% (equally weighted)
All projects will be graded out of 100 points, but depending upon their relative difficulty, which can’t be predicted in advance, they will not be weighted equally. The relative project weights will be given to you near the end of the semester.

Quizzes will be given in discussion section and will cover discussion and lecture material. They may be preannounced, or they may be unannounced pop quizzes.

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.

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. However, since the the grade distribution won’t be known until all coursework is graded, whether there is a curve, or what the letter grade ranges might be, can’t be predicted before then.

7 Exam and final dates

Midterm exams will be held during the lecture time. The final exam date appearing below is fixed however, and 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. (The only students whose finals are at the same time as this course’s final should be those who are also taking BIOM 301, BMGT 350, or EDMS 451.) If either of these situations applies to you, you must inform the instructor by the drop date this semester for allowances to be made. Also please let the instructor know immediately if you have a conflict with any of the tentative midterm dates, or any other important date as the semester progresses.

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, and announced in class.

Exam #1: Thursday, February 26 (during the lecture time)
Exam #2: Tuesday, April 21 (during the lecture time)
Final exam: Thursday, May 14, 4:00-6:00 p.m.

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. Attempts to submit a project using any other means (such as an emailed project) will not 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, and to check a project’s results on the public tests (see below) on the submit server after submitting it.

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 ten-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 will report scores for them, but we will not give them any credit in the grades we have recorded.

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 just before 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 and 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 a set of public tests will be made available on the project submission server, and each submission’s results on the public tests will be visible on the submit server soon after making the submission. Projects will also be run on various other test cases that will not be provided in advance (private or secret tests); these will test conditions the public tests do not. This course will not use any release tests, which you may have had in the preceding courses. Projects will be graded out of 100 points as follows:
A project that was not submitted will receive a score of zero.

The ten-point late penalty will be applied for each day late the project was submitted, up to a maximum 30-point deduction.

The project’s score will then be computed as follows:

<table>
<thead>
<tr>
<th>Component</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>results of public and secret tests</td>
<td>85</td>
</tr>
<tr>
<td>program style and documentation</td>
<td>15</td>
</tr>
</tbody>
</table>

All credit for any test case 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 secret tests for around 35 points (with style and documentation counting for 15 points as previously 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 submissions and the highest of their scores is what you will receive.

If you submit more than once for a project, we will grade the source code of whichever submission has the highest score on the public and secret tests (also taking the late penalty into account), which may not be the last submission you make. Therefore, to maximize your grade, you should use good programming style from the beginning of your coding, rather than going back to clean up and comment your program as your last step.

Note: this is a programming course that is a required part of the curriculum for CMSC majors and minors and for ENCE majors, and that teaches language details and concepts essential for later CMSC courses. As a result, the ability to submit reasonably successful versions of the projects is necessary. Therefore a student will not be able to pass the course with a grade of C– or higher unless each of his or her projects works correctly on a majority (half or more) of the public tests for that project assignment. Even if the three-day late deadline for a project has passed, if your project does not yet work correctly on a majority of the public tests you must still submit a version which does in order to be able to pass the course, although you will not receive credit for it. If you have any questions, 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 that prints the desired output instead of computing it, or a program which works only because it takes advantage of properties that 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 will be tested and must work correctly using the gcc C compiler on the Grace cluster, using the compilation method and compiler options that 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 systems and run on the submit server, so having a working version on another system at any other time (or even another working version in your Glue account) cannot be considered in grading. 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 that may arise when transferring it to the Grace machines.

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. Policies that may be relevant appear in the Undergraduate Catalog, which may be reached at the following link:

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 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 you. In cases of illness simply being seen by a health professional is insufficient—medical documentation must state that you were 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 or quiz; an illness 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, or as early as feasible, and if a known conflict exists with one of the planned midterm dates appearing above, notice must be given prior to the end of the schedule adjustment period. Excused absences for quizzes will also be given in case of religious obligation, provided sufficient advance notice is given.

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 at the end of the semester. 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, begin promptly, and work diligently. 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 cannot 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 and resources at www.nethics.umd.edu/aup

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, only 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 the 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 already submitted versions of that project assignment which satisfy a majority of the public tests for that project (i.e., none of the students involved are still working on or still have to submit the project), and only if that project will not be extended upon in a later project assignment.

11 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.

12 Copyright

All course materials are copyright Larry Herman (and other CMSC faculty and instructors) © 2009. 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.