1 Prerequisites and description

<table>
<thead>
<tr>
<th>Prerequisite</th>
<th>Corequisite</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or better in CMSC 132</td>
<td>CMSC 250 (or equivalent)</td>
<td>4 credits</td>
</tr>
</tbody>
</table>

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

The course’s primary objectives are to develop a good working knowledge of how to program in C, to understand how to write programs with explicit memory allocation and deallocation, to understand the UNIX and C program memory model as well as what happens when a program is running, and to gain an idea of basic systems programming concepts.

2 Contact information

2.1 Email contact

Due to time constraints and other factors it is not practical to provide detailed information or assistance regarding programming projects via email, and attempting to do so often results in students receiving incomplete or inadequate information. Therefore please ask questions about projects in person, either during office hours, or before or after class.

2.2 Instructor

Name: Derek Juba
Email: juba@cs.umd.edu

3 Class webpage

Many course materials will be made available via the class webpage at www.cs.umd.edu/class/summer2010/cmsc212. Accessing the webpage will require an ID and password to be provided in class. Students will need to check the webpage often, especially near projects due dates, because important clarifications or corrections may be made there.

4 Textbooks


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

5 Course topics (SUBJECT TO CHANGE)

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

1. Introduction: moving from Java to C (3 weeks)
2. Pointers and memory management (1 1/2 weeks)
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 1/2 weeks)
6 Computing environment and project submission system

Students will use their own TerpConnect (previously called Glue) accounts to access OIT’s Grace UNIX Cluster to do coursework. If you do not have a TerpConnect account already, request one immediately at www.oit.umd.edu/new. Projects will be turned in using the same CMSC project submission and testing server used in the preceding courses at https://submit.cs.umd.edu. However, a different mechanism will be used to turn projects in to the server in this course; details will be provided in discussion section. Unlike the previous courses, the program development environment will not be the Eclipse IDE, but rather command-line use of Linux.

6.1 Computing resources

You may use any other available system (such as your own computer) to develop coursework, although some projects may have files that will only work on an architecture identical to the Linux Grace hosts. However, all project submissions must work correctly using the gcc C compiler on the Linux Grace hosts, using the compilation procedure and options to be specified. Because different C compilers or different versions of gcc may be installed elsewhere, a program may work perfectly on one system, yet not on the Linux Grace machines. Programs will be graded based on their results on the Linux Grace systems and run on the submit server, so having a working version on the same or any other system that was not the one submitted cannot be considered in grading, nor can allowances be made for errors made in transferring files. If you want to write a project on another system you are strongly recommended to complete it several days early, to have time to address any problems related to file transfers or language or system incompatibilities, and it is strongly recommended that you use a Linux or UNIX machine to develop projects.

7 Attendance and grades

Grades will be recorded on the CMSC lower-level course grade server at https://grades.cs.umd.edu and may be checked for correctness there.

Students are responsible for all academic and administrative material 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 miss class for various reasons, but email and office hours are not intended as a replacement for class attendance. A student who was not class will need to find out what was missed and to get notes from a classmate who was present.

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

- Midterm: one midterm 25%
- Final: will be comprehensive 30%
- Programming projects: six expected coding assignments 30%
- Quizzes: in discussion section 15% (equally weighted)

The project submission and grading policies will be provided when the first project is assigned.

All projects will be graded out of 100 points, but depending upon their relative difficulty, which can’t be predicted in advance, they won’t be weighted equally. Their weights will be given at the end of the semester.

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 in advance.

8 Quiz, exam, and final dates

Quizzes will be given in discussion section and will cover discussion and lecture material. Midterm and final exams will be held during lecture. The dates will be confirmed later, and may vary depending on lecture progress and other factors.

<table>
<thead>
<tr>
<th>Midterm exam</th>
<th>Final exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friday, July 2</td>
<td>Friday, August 6</td>
</tr>
</tbody>
</table>

9 Absences and accommodations

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, at 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.
9.1 Excused absences

Missing an in–class assessment (a quiz or an exam) for reasons outside of your control (such as illness, religious observance, participation in required university activities, or family or personal emergency such as a close relative’s funeral or serious accident) will be considered to be an excused absence. Students requesting an excused absence must furnish documentary support of the cause of the absence. For an absence due to medical reasons documentation would be from a health care professional who treated you. Excused absences will not be given unless documentation is provided.

In cases of illness the documentation must show that you were treated by a health professional, and that in their judgment you were incapacitated and therefore unable to attend, for an absence to be considered excused. The documentation must include the phone number of the health professional, and indicate the exact dates or times of incapacitation, which must include the date of the missed assessment.

It is the University’s policy to provide accommodations for students with religious observances conflicting with in–class assessments, but it is the student’s responsibility to inform the instructor in advance of intended religious observances that will (or may) conflict.

An excused absence for an exam will be handled either by averaging the student’s scores for the other exams (possibly a weighted average), or by giving a makeup exam. In the case of an excused absence for a quiz no makeup will be given; the score will be computed as the average of the student’s scores for the other quizzes at the end of the semester.

A student who might miss an in–class assessment for any reason other than circumstances outside of their control must contact the instructor in advance (or as soon as possible) to discuss the reason. According to the University policy, an instructor is not obliged to make allowances other than for reasons such as those discussed above.

The policies for excused absences above do not apply to project assignments. In cases of extremely serious, lengthy, documented illness or other protracted, severe emergency situations, the instructor may consider extensions on project assignments, depending upon the 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.

Programming projects are to be written individually, so 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 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.

- In learning the course concepts 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 debugging or syntax errors. Questions like “What happens in C if you include the same header file more than once?” or “What does ‘ISO C90 forbids mixed declarations and code’ mean?” would be allowable to discuss with others because they convey no information about the contents of a student’s program.

- When it comes to actually designing, coding, or debugging 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 below, is a violation of the code of academic integrity and will be reported to the Honor Council. Questions such as “How did you implement this part of the project?” or “Please look at my code and help me find my stupid error!” exemplify disallowed cooperation.

Violations of the Code of Academic Integrity may include, but are not limited to:

1. Failing to do 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 anyone else’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.
When designing or writing projects students are free to use information and code provided by the instructional staff, only if the source is cited in a comment in the relevant section of the program, and the substantial part of a student’s program remains their own individual work.

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 please see the teaching assistants in office hours - do 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.

Students are welcome and encouraged to compare or discuss with others their implementations of programming projects after they are graded, provided that none of the students in question still have to successfully submit that project assignment, and only if it has been announced that that project will not be extended upon in a later project.

11 Right to change information

Although every effort has been made to be complete and accurate, situations that arise 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 or policies in this syllabus or in other course materials.

12 Copyright

All course materials are copyright Larry Herman (and other CMSC faculty and instructors) © 2010. 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.