1 Course 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.

Students completing this course should: (a) have a good working knowledge of how to program in the C programming language, (b) understand how to write programs with explicit memory allocation and deallocation, (c) understand the UNIX and C memory model of a program as well as what happens when a program is running, and (d) have a basic understanding of how to write systems programs.

2 Textbooks


3 Instructional Assistance and Notices

Both the instructor and TA will hold office hours in AVW 1112; the specific hours will be posted on the course website and announced in class shortly after the beginning of the course. If a student cannot attend the regular office hours, special appointments can be arranged.

As a general rule, the instructional staff prefers to provide help with projects and course material in person, as opposed to via email, as a face-to-face, real-time discussion is much better suited to helping students understand concepts and debugging programs. While we will often make an attempt to answer questions raised in email, please understand that we may have to direct you to ask the question in person during office hours instead so that we can adequately answer your question; specifically, debugging questions will not be answered via email.

Please also note that we will use the course website to broadcast information to all students, such as announcement of project details, corrections to projects, test dates, and other important information. Students are expected to check the website at least three times per week for updated information.

4 Course Topics

The following list of topics may vary according to the pace of lecture, and so is subject to change. All times listed are approximate.

- Introduction: moving from Java to C (2 weeks)
- Pointers, memory management, and dynamic data structures (2 1/2 weeks)
- I/O, standard libraries (1 week)
- Testing (1/2 week)
- Data representation (1/2 week)
- Libraries and linking (1/2 week)
- Process control (1 week)
- Program measurement and optimization (1/2 week)
- Data representation and implementation of functions (1 1/2 weeks)
5 Course Grading Policies

Final course grades will be determined using the following percentages:

- **Midterms:** two midterms, equally weighted 30%
- **Final exam:** will be comprehensive 25%
- **Programming projects:** approximately six coding assignments 35%
- **Quizzes:** in discussion section 10%

Quizzes will cover discussion and lecture material; they may be preannounced or unannounced pop quizzes.

Projects must be submitted via the submit server, by the deadline indicated by the assignment. Late projects will be accepted up to 24 hours late, with a penalty of 25 points deducted for all late submissions. Submission deadlines are enforced by the submit server, and exceptions to these deadlines can be made only in extreme situations – specifically, system problems, network problems, and power outages are not cause for the instructor to extend a deadline, and so students are advised to begin work early on projects and submit programs as early as possible.

Projects will be graded using a combination of automated testing with the submission server, and source code grading. The automated testing will consist of public and secret tests – please note that unlike previous courses, release tests will not be used.

If both an on-time and a late submission is made, the late penalty will be applied to the late submission and the higher of the two scores will be used. If multiple submissions are made in a category (on-time or late), the last submission in a particular category which compiles will have its results used as the representative of that category.

Source code grading will also form a portion of your project grades; we will select only one of your submissions for this grading. The submission selected will be the last on-time submission, or the first late submission if no on-time submissions are made.

Understanding programming concepts is a hands-on activity, and so all students are expected to complete and submit all programming projects. The instructor reserves the right to fail, regardless of overall numeric score, any student who does not submit a good-faith effort to complete all projects. This means that, even if the late deadline has passed for a project, you must still submit a reasonably complete version of the project in order to be able to pass the course, although you will not receive any score for it.

Grades will be recorded and may be checked for correctness on the CMSC grade server, accessible at http://grades.cs.umd.edu

6 Computing Environment and Project Submission

Programming will be done on the OIT Grace cluster, in particular on the two Linux machines that can be accessed via SSH at linux.grace.umd.edu. Students will use their own Glue (also known as TerpConnect) accounts to access the Grace cluster and do coursework, so students who do not yet have a Glue account should request one online immediately by visiting http://www.oit.umd.edu/new

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

However, students will use a different mechanism to turn projects in than was used in preceding courses; details will be provided with the first project assignment.

While students may use other environments to develop their programs, to be judged correct, a project must build and work correctly on the Linux Grace systems, with the compilation method and compiler options specified for that project.

7 Attendance and Absence Policies

This course follows the University policies on attendance and excused absences, as specified in the Undergraduate Catalog, which you may view at http://www.umd.edu/catalog

Documentation for absences due to medical reasons must contain a statement that you were incapacitated, the phone number of the health care professional who examined you, and the dates of incapacitation (which must include the dates of the missed exam or quiz).
It is the student’s responsibility to inform the instructor of any expected excused absences ahead of time. For exams, students are expected to inform the instructor of a conflict in writing (email is acceptable) as soon the exam is announced or the conflict is known, whichever occurs first.

There will be no makeup quizzes given; the score for a quiz missed due to an excused absence will be omitted in determining the average of the student’s scores during final grade calculation. Exams missed due to an excused absence will be dealt with by the instructor on a case-by-case basis, either with a makeup exam or by altering the weights of the other exams the student has taken while calculating the final grade.

An excused absence does not relieve the student of the obligation to turn in programming projects on time, as projects are assigned well in advance of their due dates. In cases of a lengthy illness, or other protracted emergency situations, the instructor may consider extensions on project assignments, depending on the specific circumstances.

8 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, within the first week of classes. All arrangements for exam accommodations as a result of a disability must be made and arranged with the instructor at least three business days prior to the exam date, or accommodations can not be made.

9 Academic Integrity

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.

You may review the University’s Code of Academic Integrity for yourself at http://www.shc.umd.edu/code.html

Please carefully read the Office of Information Technology’s policy regarding acceptable use of computer accounts and resources at http://www.nethics.umd.edu/aup

Unless stated otherwise by the instructor, all programming assignments are to be written individually. Cooperation between students on exams, quizzes, or projects is a violation of the Code of Academic Integrity. Any evidence that a violation of the Code has occurred will be submitted to the Student Honor Council, which could result in an XF for the course, suspension, or expulsion. Automated tools may be used to compare students’ code to look for evidence of cheating.

Students are welcome and encouraged to study and compare or discuss their implementations of the programming projects with others after they are graded. However, before a project’s results are announced, students should not discuss or examine each other’s solutions for that project in any way. If you have any question about the appropriateness of a particular situation then consult with the instructors in advance. Should you have difficulty with a programming assignment you should see the instructional staff in office hours, NOT solicit help from anyone else in violation of these rules.

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

10 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.

11 Copyright

All course materials are copyright Derrick Wood (and other CMSC faculty and instructors) © 2009. All rights reserved. Students are permitted to use course materials for personal use only. Course materials may not be distributed publicly or provided to others (excepting other students in this course), in any way or format.