1 Official course description and overview

Prerequisite: permission of department. Pre- or corequisite: MATH 140. Only for CMPS, ENCP and students with major code: 2299F. Not open to students who have completed CMSC 114 or higher. Design and analysis of programs in C. An introduction to computing using structured programming concepts. For further information contact the Undergraduate Education Office, Computer Science Department.

CMSC 106 is an introduction to computing and program development in the C programming language. This includes a brief introduction to basic computer concepts, an understanding of the UNIX operating system sufficient for writing class programming assignments, studying the syntax and semantics of the basic control structures of C, learning C’s fundamental data types and structures, understanding the design and methodical construction of computer programs, learning how to test and debug programs, and lastly, practice in these through creating several programs in C. The major topics to be covered can be found in the last section of this syllabus.

2 Class webpage

Various course materials will be made available on the class webpage, and any important announcements will also be made there, so students are responsible for visiting the class webpage frequently. The class webpage is at the following URL: www.cs.umd.edu/class/fall2006/cmsc106. Accessing the web page requires an ID and password which will be provided in class.

3 Textbook


All of the example programs, and answers to the exercises in the text, can be downloaded from the publisher’s website (URL to be given in the class webpage), and if needed versions of these programs formatted for the UNIX system will also be provided.

4 Contact and office hours information

4.1 Email contact

Although our email addresses are provided below, we will generally be unable to provide long explanations about the course material via email; these are more appropriate for class discussion or personal communication. In particular, most types of detailed questions about the programming projects usually cannot be answered adequately via email. Personal communication, either in office hours or before or after class (as time permits), is the preferred means of contacting us, as we will always be able to devote more attention and to give a much more detailed reply to any issue which is discussed in person. In particular, on days when class is held, please try to bring questions to class to ask, or before or after class, rather than using email.

4.2 Instructor

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
<th>Office hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Larry Herman</td>
<td>(301) 405–2762</td>
<td>M 1–2:30, Th 2:30–3:30</td>
</tr>
<tr>
<td>1111 A. V. Williams</td>
<td></td>
<td>(or by appointment)</td>
</tr>
<tr>
<td><a href="mailto:larry@cs.umd.edu">larry@cs.umd.edu</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.3 Teaching assistants

<table>
<thead>
<tr>
<th>name</th>
<th>duties</th>
<th>email</th>
<th>office hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nick Frangiadakis</td>
<td>teaching, 0101 &amp; 0102</td>
<td>.ntg (*)</td>
<td>M 11:30–1, Th 1–2, F 12:30–2</td>
</tr>
<tr>
<td>Walaa El-Din Moustafa</td>
<td>teaching, 0103</td>
<td>walaa (*)</td>
<td>Tu 11:30–12:30, W 12:30–2:30, Th 11:30–12:30</td>
</tr>
<tr>
<td>Thuan Huynh</td>
<td>grading</td>
<td>thuan (*)</td>
<td>Tu 2:30–4:30, W 11–12, F 3–4</td>
</tr>
</tbody>
</table>

(*) All of the TAs’ email addresses are “@cs.umd.edu”.

4.4 Office hours assistance with projects

All the TAs’ office hours will be held in 1112 A.V. Williams, and the TAs’ office hours will be provided separately shortly. Note that assistance with and suggestions regarding project problems are provided during office hours, but it is the responsibility of each student to develop and to do most of the debugging of his or her own program. You should not expect the instructional staff to take the primary responsibility for getting your project to work.

5 Course requirements and grading policies

Students are responsible for all material covered and announcements, policies, and deadlines discussed in lecture and discussion section, even if they were not in class to hear the information.

Practice exercises and solutions will be made available via the class webpage; these will not be collected or graded, but allow you to test your knowledge of the material in preparation for exams, quizzes, and projects. Read the relevant sections and work on the these exercises when topics are first covered to insure you are prepared when it is time to write the programming assignment requiring this material. If you have trouble with the problems, be sure to come to office hours for assistance.

Quizzes will be given during the discussion or lab section and will be announced in an earlier class.

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

- Quizzes: on reading, lecture, and discussion material 10%
- Programming projects: approximately six coding assignments 30%
- Midterms: three midterms are expected 40% (equally weighted)
- Final: will be comprehensive 20%

Any request for reconsideration of the grading on any coursework must be submitted within one week of its return, or it can not be considered. Exam regrading requests must be made in writing. Information about resolving any questions about project grading will appear in each graded project. The instructional staff may regrade the entirety of any exam or project submitted for regrading.

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.

Dates related to exams are:

- Exam #1: (Tentative) Wednesday, October 4
- Exam #2: (Tentative) Wednesday, November 8
- Exam #3: (Tentative) Monday, December 4
- Final exam: Friday, December 15, 4:00–6:00 p.m.

The exact dates for the midterms will be confirmed later, and may vary depending on lecture progress and other factors (you will be told in advance). The final exam date is fixed by the University. The final exam will be rescheduled only for students having another final at exactly the same time, or for students with more than three exams scheduled on the same day. (The only students whose finals are at the same time as this course’s final are those also taking BMGT 221 or ENES 221.) 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. Please also 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.
6 Computing resources, project submission, and grading policies

6.1 Computing resources

Students will use their own Glue accounts to access the OIT Grace UNIX Cluster for coursework to be done. If you do not have a Glue account already, request one immediately online at www.oit.umd.edu/new.

Due to issues regarding incompatibility between compilers, operating systems, file formats, etc., on different computer systems, students are strongly advised to do all of their program development on the Grace cluster. If you have access to another computer with a C compiler, such as a personal computer at home or another computer system at your job, and you choose to work on your programming projects there, you should keep the following information in mind. All project submissions must be made from your Glue account on the OIT Grace Cluster and must run and work correctly using the gcc compiler available on the Grace cluster; the exact instructions and requirements will be provided later. Due to implementation-dependent behavior permitted by the C language standard, sometimes a program can work perfectly when compiled with one compiler or on one system, yet not work right when compiled with another compiler or elsewhere. If the program you submit does not work correctly on the Grace cluster that is how it will have to be graded, regardless of whether you had a working version anywhere at an earlier time.

The instructional staff cannot provide full assistance with any program during office hours unless it is in your Glue account. Consideration in grading cannot be made for errors made in transferring files, including transferring or submitting the wrong file. 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 related to file transfers or language or system incompatibilities.

6.2 Project submission

Projects will be submitted electronically and further instructions will appear on the assignments and be covered in discussion section. Attempts to submit a project using any means other than that specified on the assignment, such as via email, can not be considered. Only the projects electronically submitted from a student’s Glue account, according to the procedures provided, 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 exactly at 10:00:00 p.m. on the day indicated on the project assignment. A project will be accepted up to three days late with a 10% deduction in credit for each day. The submission deadlines are firm and exceptions cannot be made. Note there is no grace period for project submissions—deadlines will be enforced at exactly the times indicated. The system time on the OIT Grace UNIX Cluster will govern all submissions, and may be checked on the Cluster hosts at any moment using the UNIX date command.

6.3 Project grading

All projects will be graded out of 100 points, but as the semester progresses they will be larger, and consequently will count more towards the final grade (in other words, the projects will not all be weighted equally). The relative weights of the projects can’t be given in advance, but will be given toward the end of the semester.

Projects will be graded as follows. All projects will be run on a primary input data set which will be made available with each project assignment. The program’s results when run on the primary input will also be given, which are called the primary output. Projects will also be run on various secondary inputs which will not be provided in advance. A submitted program is considered to satisfy an input data set if, when run on that input, all the data values in each line of its output exactly match the data values in the corresponding lines of the correct output (and extra incorrect data values are not produced also). A project’s grade will be calculated in the following manner:

- Any project which fails to compile will receive a grade no higher than 40 (less any late penalty).
- Any project which is a reasonable attempt to solve the assigned problem which compiles and runs but does not satisfy the primary input will receive a grade no higher than 50 (less any late penalty).
Any project which satisfies the primary input will be graded as follows:

- correct results for primary input: 50 points
- results for secondary inputs: 30 points (prorated)
- style and documentation: 20 points (prorated)

Any late penalties will also apply to such projects.

Projects not submitted by the end of the three-day grade period will not receive any credit, so submissions for projects after the three-day period will not be considered. As described above, all projects submitted will receive at least some partial credit, even those which do not produce correct results, so it is to your advantage to submit something for each project before the end of the third day, even if it is incorrect or incomplete.

If more than one submission has been made by a student for any project assignment (for example, both an on-time submission as well as a two-day late submission) all submissions will be graded and the highest grade will be recorded. Only the last submission of each type will be graded, so for example any on-time submission made will overwrite or replace an earlier on-time submission by that student for that project, and any one-day late submission made will overwrite or replace an earlier submission by that student for that project also made within the 24-hour one-day late period.

### 7 Late policies, and accommodations and 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.

It is the University’s policy to provide reasonable accommodations for students observing religious holidays which conflict with exams, but it is the student’s responsibility to inform the instructor in advance of any intended religious observances. Written notice must be provided immediately upon an exam date being announced or confirmed in order for accommodation to be made.

There will be no makeups for missed quizzes – in the case of a reason such as religious observation, documented medical excuse, or documented family or personal emergency, (such as a serious automobile accident or close relative’s funeral) the score for a missed quiz will be counted by averaging the student’s scores for the other quizzes. The score for an excused absence for an exam will at the instructor’s discretion be determined either by averaging a student’s 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, even in the case of one of the causes mentioned where proper documentation is available. If you might miss an exam for any reason other than those mentioned above (such as attendance at a required University function), you 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.

Students claiming a excused absence must apply in writing and furnish documentary support (including the phone number of a contact person) for any 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. Simply being seen by a health professional is insufficient – medical documentation must state that you were incapacitated for an excused absence to be considered. For medical absences the documentation must explicitly indicate the exact dates or times of incapacitation due to illness. Self-documentation of illness is not itself sufficient support to excuse an absence. For medical absences, the dates of incapacitation on the documentation must include the date of the missed exam or quiz; an illness preceding an exam or quiz will not be considered to be an excused 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 any medical documentation.
Projects will be assigned with sufficient time to be completed by students who have a reasonable understanding of the necessary material and who begin promptly. In cases of extremely serious illness of lengthy duration (such as, for example, an illness requiring hospitalization), the instructor may consider extensions on project assignments, depending upon the specific circumstances.

7.1 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.

8 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.

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

- 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 features of C, what was discussed in class, and general syntax errors.
- When it comes to actually writing a project assignment, other than help from the instructional staff it 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, will be considered a violation and WILL BE REPORTED to the Honor Council. Students may not discuss design of any part of a program which is to be submitted with anyone except the instructional staff. Students may not use any disallowed source of information as a basis for or in creating either their project design or code.

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 instructional staff in office hours, NOT solicit help from anyone else in violation of these rules.

Students are welcome and encouraged to study and compare their implementations of the programming projects with others after the particular project has been graded and returned.

Violations of academic honesty 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.
It is the responsibility, under the university’s honor policy, of any student who learns of an incident of academic dishonesty to report it to their instructor.

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

9 Course topics (SUBJECT TO CHANGE)

This chart gives the estimated time for each chapter of the text, but in a few cases coverage of material will deviate from the exact order of the chapters.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Chapter</th>
<th>Approximate time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course introduction, and introduction to computer programming</td>
<td>Chapter 1</td>
<td>1 lecture</td>
</tr>
<tr>
<td>Getting started in C programming: introduction to basic program syntax and printing messages, data types and declarations, numeric and character data, expressions, printing results, and variables</td>
<td>Chapter 2</td>
<td>3 lectures</td>
</tr>
<tr>
<td>Processing and interactive input: assignments, counting, and input and output of data</td>
<td>Chapter 3</td>
<td>1 lecture</td>
</tr>
<tr>
<td>Selection: C’s conditional statements</td>
<td>Chapter 4</td>
<td>4 lectures</td>
</tr>
<tr>
<td>Repetition: C’s looping statements</td>
<td>Chapter 5</td>
<td>4 lectures</td>
</tr>
<tr>
<td>Modularity using functions: user-defined functions, parameters and return values, standard library functions, scope, call-by-reference</td>
<td>Chapter 6</td>
<td>2 lectures</td>
</tr>
<tr>
<td>More modularity using functions: scope, storage class, and call-by-reference</td>
<td>Chapter 7</td>
<td>2 lectures</td>
</tr>
<tr>
<td>Arrays: single and two-dimensional arrays, initializers, array parameters</td>
<td>Chapter 8</td>
<td>3 lectures</td>
</tr>
<tr>
<td>Character strings: processing strings using loops, some character and string library functions</td>
<td>Chapter 9</td>
<td>2 lectures</td>
</tr>
<tr>
<td>Arrays, addresses and pointers: the relationship between arrays and pointers</td>
<td>Chapter 11 (partial)</td>
<td>1 lecture</td>
</tr>
<tr>
<td>Structures: structure concepts, structures as parameters, arrays of structures</td>
<td>Chapter 12 (partial)</td>
<td>3 lectures</td>
</tr>
<tr>
<td>Additional or miscellaneous topics</td>
<td>other material as time permits</td>
<td>3 lectures</td>
</tr>
</tbody>
</table>

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 Larry Herman © 2006. 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.