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 Section 11.

2 Contact and office hours information

2.1 Email contact

Although our email addresses are provided below, we will generally be unable to provide lengthy explanation about the course material via email; this is more appropriate for class discussion or personal communication. Most types of questions about programming projects cannot usually 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 discussed in person. On days when class is held, please try to bring questions in person to class.

2.2 Staff

<table>
<thead>
<tr>
<th>Name</th>
<th>Larry Herman</th>
<th>Sanjiv Dinakar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone:</td>
<td>(301) 405–2762</td>
<td>N/A</td>
</tr>
<tr>
<td>Office:</td>
<td>1111 A.V. Williams</td>
<td>1112 A.V. Williams</td>
</tr>
<tr>
<td>Email:</td>
<td><a href="mailto:larry@cs.umd.edu">larry@cs.umd.edu</a> (see Section 2.1 above)</td>
<td><a href="mailto:sdinakar@cs.umd.edu">sdinakar@cs.umd.edu</a> (see Section 2.1 above)</td>
</tr>
</tbody>
</table>

2.3 Office hours, and assistance with projects

Office hours will be provided in a separate handout shortly.

Assistance with and suggestions regarding project problems are provided during office hours, but it is ultimately the obligation 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 projects to work.

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 supplied on the class webpage), and versions of these programs formatted for the UNIX system will also be provided as needed.

4 Class webpage

Certain course materials will be made available on the class webpage, and some important announcements may be made there, so students are expected to check the webpage frequently. It can be accessed at the link
5 Course evaluations

Course evaluations are important, and the department and faculty and instructors take student feedback seriously. The evaluation system for this semester will be open between between Tuesday, December 1, and Sunday, December 13, at www.courseevalum.umd.edu. However, and more importantly, rather than waiting until the end of the semester to give feedback, please bring any questions or concerns to my attention during the course; preferably in person if at all possible. An instructor cannot guarantee to be able to solve any problem or change anything that is discussed, but can’t do anything if any issues that arise are not brought to their attention, so I welcome hearing any comments, questions, suggestions, or concerns that you may have.

6 Computing environment and resources

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.

Due to incompatibilities between 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 one at home or 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 run and work correctly using the gcc compiler available on the Grace cluster. 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 be graded, regardless of whether you had a working version anywhere at an earlier time. Consideration in grading cannot be made for errors made in transferring files to or from the Grace systems.

The instructional staff cannot provide full assistance during office hours with any program unless it is on the OIT TerpConnect/Grace systems, in your extra course disk space (to be explained later). 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.

7 Attendance and graded materials

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. 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 is expected to find out what was missed and get notes from a classmate who was present.

Practice exercises and solutions may periodically be made available via the class webpage; these will not be collected or graded, but will be for the purpose of allowing 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.

Graded coursework will count towards the final grade according to the following percentages:

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

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

The midterm dates will be confirmed later, and may vary depending on lecture progress and other factors. The final exam date is fixed by the University, and will be rescheduled only for students having another final at exactly the same time, or for students with more than three finals 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 at least two weeks in advance of the final exam time for any allowances to be made. Also please inform the instructor immediately if you have a conflict with any of the midterm dates, or any other important date as the semester progresses.

Exam #1: Monday, October 5
Exam #2: Wednesday, November 4
Exam #3: Wednesday, December 2
Final exam: Tuesday, December 15, 4:00–6:00 p.m.

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 judgement 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. Self–documentation of illness will not be accepted. 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 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 makeups 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. 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 lasting longer than a week 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 features of C, what was discussed in class, and general debugging or syntax errors.

- 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, 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.

Violations of academic honesty 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.

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—do 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.
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.

Students are welcome and encouraged to compare or discuss with others their implementations of programming projects after they are graded, provided that it has been announced that that project will not be extended upon in a later project.

11 Course topics (subject to change)

This table 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>Exams #1</td>
<td></td>
<td></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>Exams #2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arrays: single and two–dimensional arrays, initializers, array parameters</td>
<td>Chapter 8</td>
<td>2 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>2 lectures</td>
</tr>
<tr>
<td>Extra work</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Additional or miscellaneous topics</td>
<td>other material as time permits</td>
<td>2 lectures</td>
</tr>
</tbody>
</table>

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 © 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.