

CS 109, C/C++ Programming for Engineers with MATLAB

Course Policies - Spring 2012 - **UPDATED**

Course Objectives: At the end of this course, students will be able to:

- Apply engineering problem-solving techniques to the solution of engineering problems.
- Develop, write, test, and debug simple computer programs in C/C++ for solving engineering problems.
- Develop logical thought patterns for implementing engineering solutions, including linear, branching (conditional), and repetitive execution, and to implement these concepts in C/C++ programs.
- Develop, write, test, and debug simple MATLAB programs for the solution of engineering problems, including the graphing of data generated in either MATLAB or from other sources.

Instructor:

John Bell
jbell@cs.uic.edu
<http://www.cs.uic.edu/~jbell>
1035 SEO, 413-9054
Office Hours: May be M-Th 1:00 – 2:00
See web for details
Open Door policy during other times.

Teaching Assistants:

Sihong (Shawn) Xie, sxie6@uic.edu – **Thur 8, 9, 11:00 sections.**
Kaiser Asif, kasif2@uic.edu – **Fri 8, 9:00 sections.**
Gurudev Devanla, gdevan2@uic.edu – **Thur, Fri 10:00 sections.**
See web site for office hours and locations

Corerequisite: Math 180

If you are an undergraduate student who does not have the necessary pre-requisites, DROP THE CLASS NOW. Otherwise you will be automatically dropped later, when it will be too late to sign up for anything else instead.

Credits:

3

Course Web Page:

<http://www.cs.uic.edu/~i109>

Lab:

SEL 2249, except 2249F for Thursday 9:00 and SCE408 for Thursday at 10:00 – Check web site & e-mail for confirmation.
Labs start the **FIRST WEEK** of classes.

Textbooks:

Required:

- Pearson Custom "Computer Science - CS 109 C/C++ with Matlab", ISBN-13: 978125603872-6. This is a custom-prepared book, with chapters excerpted from:
 - Delores M. Etter and Jeanine A. Ingber, "Engineering Problem Solving with C++, Second Edition", Prentice Hall, ISBN-13: 9780136011750.
 - Delores M. Etter, "Introduction to MATLAB", Second Edition, Pearson Prentice Hall.

Other Recommendations:

- Brian W. Kernighan and Dennis M. Ritchie, "The C Programming Language", 2nd Ed.
- Ray Lischner, "C++ in a Nutshell", O'Reilly, 2003.
- Kyle Loudon, "C++ Pocket Reference", O'Reilly, 2003.
- Steve Oualine, "Practical C++ Programming, 2nd Edition", O'Reilly, 2002.
- <http://www.cplusplus.com>
- Hahn and Valentine, "Essential MATLAB for Engineers and Scientists", 4th Edition.

CodeLab:

During this term, we will be using an automated on-line exercise program called CodeLab, which will give you immediate feedback on your work and show a variety of different acceptable solutions to any problem in addition to other features and benefits. **For this semester, Codelab will be semi-optional, strongly recommended for new programmers.**

REGISTRATION:

1. Go to www.tcgo1.com --OR-- www.tcgo2.com
2. Click "Register for CodeLab"
3. choose "I am a student in a course ..." and click CONTINUE
4. enter the Section Access Code: **ILLIILLI-7230-HVYB-11** and click CONTINUE
5. continue filling out the forms being careful to enter a VALID email address and first and last names (these will appear in the professor's roster)

LOGIN:

1. Go to www.tcgo1.com --OR-- www.tcgo2.com
2. Click "Login to CodeLab"
3. the username and password are the email and password given during registration.
4. A \$25 fee is required to get full access to CodeLab. (Go to the "Lobby" link.)

Academic Integrity:

Students are encouraged to study together and to help each other learn. When one student teaches another, both benefit from the experience. However, it is a strict violation of class and university policy for any student to hand in any work that is not 100% their own creation. Therefore:

- All work on all exams and all homeworks must be individually performed by the student whose name appears on the paper.
- No student may give any other student any portion of their code, either written down, electronically, or through any other means.
- Students are responsible for safeguarding the integrity of their work. This includes but is not limited to changing their passwords and keeping their computer accounts secure.
- Direct copying of code from any textbook or other source is strictly forbidden.
- Students may discuss homework problems, including background concepts and general solution strategies, but they are forbidden from discussing or sharing specific solutions. **In particular, it is forbidden for any student to show any other student any portion of their computer programs or homework solutions for any reason, including debugging assistance.** This means you must hand in your own homework. **You are not allowed to see anyone else's work, or show your work to anyone. Failure to protect the privacy of your work is also a violation.**
- All submitted programs will be analyzed using MOSS, to identify any unacceptable similarity to other students' code or to previous or published solutions if applicable.
- In the case of extreme discrepancy between homework performance and exam performance (e.g. very high homework scores and very low exam scores), the instructor shall determine which scores more accurately reflect the students' true work.
- First violations will be immediately assigned a **NEGATIVE** score, **for all parties involved in the transgression**, and may also be penalized with a grade reduction and/or failure. Second or more serious violations will be reported directly to the Vice Chancellor for Student Affairs, and may result in a failing grade, probation, suspension, or expulsion from the university, as well as being documented on the permanent records of all students involved.

Planned Schedule:

The following schedule is planned, as of January 2012, and is subject to dynamic adjustment as necessary. (A more detailed schedule may appear on the web site.)

Weeks	Topic	Chapters & Sections	Notes
1-2	Course Intro, Simple C++ Programs and Operations	Syllabus, Ch 1 Sections 1-4, 7	HW1 due week 2 NO CLASS 16 JAN.
3-4	C++ Control Structures	Ch 2 Sections 1-4 Ch 12 - Appendix	QUIZ WEEK 3 HW2 due week 4(Th)
5-6	C++ File Operations	Ch 3 Sections 1-3, 5	QUIZ WEEK 5 HW3 due week 6
7	All of the above	All of the above	MIDTERM I WEDNESDAY 22 FEB 6-8 P.M. (see below)
7-8	C++ Functions	Ch 4 Sections 1-3	QUIZ WEEK 8
9-10	C+ Arrays (2D and multi-D)	Ch 5 Sections 1,7,9 Ch 6 Section 1, 6	HW4 due week 9 QUIZ WEEK 10
	SPRING BREAK MARCH 19-23		
11	All of the above	All of the above	HW5 due week 11(M) MIDTERM II THURSDAY 29 MARCH 6 - 8 P.M. (see below)
11	Matlab - Getting Started	Ch 7	
12	Matlab - Using and Writing Functions	Ch 8	QUIZ WEEK 12
13	Matlab - Plotting	Ch 9	HW6 due week 13
14	Matlab - Control	Ch 10	QUIZ WEEK 14
15	Matlab - Matrices	Ch 11	HW7 due week 15
Finals	All of the Above	All of the Above	Final Exam WEDNESDAY 2 MAY 1:00 – 3:00 P.M.

MIDTERM EXAM I, WEDNESDAY 22 FEBRUARY, 6:00 – 8:00 P.M.

MIDTERM EXAM II, THURSDAY 29 MARCH, 6:00 – 8:00 P.M.

FINAL EXAM WEDNESDAY 2 MAY 1:00 – 3:00 P.M.

QUIZZES IN LAB WEEKS 3, 5, 8, 12, AND 14.

HW due 8:00 a.m. on Wednesdays. Solutions presented in lab starting 8:00 Thursday.

(Exceptions: HW2 due Thursday, HW5 due Monday with no late submissions.)

Mandatory Laboratory Attendance:

Attendance at all laboratory sessions after the initial add/drop period is required. If you are an advanced student who feels you have nothing to learn from the labs, then you can spend the lab period exploring advanced concepts or else helping your fellow students - You will find that you never truly understand something until you have to explain it to someone else.

Laboratory Scores

TAs will assign a "laboratory score" to each student at the end of the semester. In addition to attendance, laboratory scores may be based on participation, effort and attitude as well as performance on lab activities. Lab scores will be independent of homework and quizzes.

Grading Policy:

Numerical scores will be based upon the following contributions:

(3) Exams (2 midterm, 1 final)	20 points each
Programming assignments	10 points each
Quizzes	5 points each
CodeLab assignments	Borderline Consideration
<u>Lab Score</u>	<u>10 points</u>
Total:	Normalized to 100 point scale

Unless otherwise specified, all programming & homework assignments will carry equal weight. The exact number of such assignments will be determined as the course progresses.

Conversion of numerical scores to letter grades is a serious business, requiring careful consideration of every student's **complete** semester performance, **and will not be considered until all scores are compiled at the end of the semester.**

There are no predetermined grade guarantees. However it is expected that grades will follow the general pattern given below. **Regardless of the numerical score, it will not be possible to pass this course without passing the exams, particularly the final exam, and completing most of the homework assignments.**

<u>The grade break for:</u>	<u>will probably be somewhere around:</u>
A / B	90
B / C	80
C / ?	70

Note that the final grade breaks may be either slightly below **or slightly above** the numbers given here.

Homework Grading Policies

Specific homework grading guidelines will be determined on a case-by-case basis. For programming assignments, it is expected that the points will break down roughly as follows:

Program compiles and runs (using Dev C++ on ACCC computers)	25%
Program handles simple, straightforward situations:	25%
Program handles more advanced and/or tricky situations:	25%
Program is efficiently written using good programming style:	25%

Notes:

1. The first 50% of any assignment should be relatively easy to get. Anyone who hands in a program that compiles, is documented, and shows a reasonable attempt to complete the assignment should get at least half the points. Scores less than 50% are for incomplete assignments or work that just isn't worth much at all.
2. Scores from 50 to 90 % are based on quality and performance, with most scores expected to be in the 70 to 90 range.
3. The last 10% should be hard to get. Scores over 90 should only go to notably excellent papers, with scores of 100 going only to perfect error-free work.

Grading Pools

It is not possible to maintain 100% consistency between all sections of this course, because different TAs will always teach and/or grade a little bit differently. Therefore separate grading pools will be established corresponding to students taught and graded by the same TA, and each student's scores will only be compared against the other students in the same grading pool. Grading between different TAs may or may not be consistent with each other, but that is unimportant, as each grading pool will be evaluated independently of the others.

Exam Policy:

- All mid-term exams will be given at night, so that students will have ample time to complete the exam.
- Exams will be written so that the average student will be expected to finish in about an hour, so time constraints will not be a factor.
- Any exam conflict needs to be brought to the instructor's attention for resolution **before** the regularly scheduled exam. Requests for make-up exams after the regularly scheduled exam will not normally be granted.
- Exams will be closed-book. One crib sheet will be allowed, no larger than 8.5x11 inches, double sided, **handwritten**.
- All exams will be cumulative, with emphasis on material which has not been covered on previous exams.
- All material covered in class or in assigned reading or which should have been learned in the course of completing homework is fair game on exams. No more specific information will be provided as to exam content.
- Anyone who fails to stop working on their exam when time is called will receive a minimum of a 5 point late penalty.

Special Considerations

- All programs must be turned in using Blackboard.
- Each program must be accompanied by a user documentation file.
- Programs must compile and run properly using Dev C++ on the ACCC XP Computers to receive full credit. **Programs which run on other systems** (e.g. Microsoft's Visual Studio) **but which do not run on the ACCC systems will be downgraded accordingly, at the grader's discretion.**
- **Programs that do not compile will automatically lose 25% of the possible points.** Non compiling programs may still be eligible for the other 50% of the points based on programming and the 25% based on documentation.
- The user documentation file is for the benefit of someone who would be using your program, but who does not get to see the code, and who has not read the assignment.
- Assignments submitted after the due date and time, but within 24 hours, will be assessed a 20% penalty. No assignments will be accepted more than 24 hours after the due date and time.
- All appeals for grading errors, no matter how justified, must be submitted within two weeks after the graded assignments are returned. No appeals for regrades will be heard after that time.