

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

## Course Policies - Fall 2009

**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 TTh 12-1:30, W2:30-3:30  
See web for details  
Open Door policy during other times.

**Teaching Assistants:**

Tathagata Dasgupta, tdasgu2@uic.edu  
Jing Wang, jwang69@uic.edu  
Xu Zhang, xzhang54@uic.edu  
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>

**Textbooks:**

**Required:**

- Delores M. Etter and Jeanine A. Ingber, "Engineering Problem Solving with C++, Second Edition", Prentice Hall, ISBN-10: 0136011756, ISBN-13: 9780136011750.
- Brian D Hahn and Dan Valentine, "Essential MATLAB for Engineers and Scientists, Third Edition", Elsevier, ISBN: 978-0-7506-8417-0, ISBN10: 0-7506-8417-8

**Other Recommendations:**

- Brian W. Kernighan and Dennis M. Ritchie, "The C Programming Language", 2<sup>nd</sup> Ed.
- Loukides, Mike and Andy Oram, "Programming with Gnu Software", O'Reilly.
- Ford, Ann R. and Toby Teorey, "Practical Debugging in C++".
- Ray Lischner, "C++ in a Nutshell", O'Reilly, 2003.
- Kyle Loudon, "C++ Pocket Reference", O'Reilly, 2003.
- Steve Oualine, "Practical C++ Programming, 2<sup>nd</sup> Edition", O'Reilly, 2002.

## 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.

### REGISTRATION:

1. Go to [www.tcco1.com](http://www.tcco1.com) --OR-- [www.tcco2.com](http://www.tcco2.com)
2. Click "Register for CodeLab"
3. choose "I am a student in a course ..." and click CONTINUE
4. enter the Section Access Code: **ILL-CHI-8821-0** 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.tcco1.com](http://www.tcco1.com) --OR-- [www.tcco2.com](http://www.tcco2.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 may be 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.
- All violations will be reported directly to the Office of Student Judicial Affairs, <http://www.uic.edu/depts/sja>. First violations will be penalized with zero on the relevant assignment(s) **and** a penalty equal to the value of the assignment(s), **for all parties involved in the transgression**. Second or more serious violations may result in a failing grade, probation, suspension, or expulsion from the university. The violation will also be recorded on the permanent records of all students involved.

**Planned Schedule:**

The following schedule is planned, as of August 2009, and is subject to dynamic adjustment as necessary. ( A more detailed schedule may appear on the web site. )

Weeks	Dates	Topic	Chapters	Notes
1 – 4	24 Aug - 18 Sept	Data types, variables, statements, and control flow	C++ 1 - 3	<b>No Class Labor Day Monday 7 Sept.</b>
<b>5</b>	<b>23 Sept.</b>	<b>All of the above</b>	<b>All of the above</b>	<b>MIDTERM I THURSDAY 24 SEP, 6-8 P.M.</b>
5-10	21 Sep - 30 Oct	File I/O, Functions, Arrays, Matrices	C++ 4 - 7	
<b>11</b>	<b>4 Nov</b>	<b>All of the above</b>	<b>All of the above</b>	<b>MIDTERM II TUESDAY 3 NOV, 6-8 PM</b>
11 - 15	2 Nov - 4 Dec	MATLAB	M 1-3, 4 to 4.2.4, 6 to 6.2, 7, 8, 10 to 10.2	<b>No Class ( Lab ) Thanksgiving Nov 26-27</b>
<b>Finals Week</b>	<b>8 December</b>	<b>All of the Above</b>	<b>All of the Above</b>	<b>Final Exam TUESDAY 3:30 – 5:30</b>

**MIDTERM EXAM I, THURSDAY 24 SEPTEMBER, 6:00 – 8:00 P.M.**

**MIDTERM EXAM II, TUESDAY 3 NOVEMBER, 6:00 – 8:00 P.M.**

**NO CLASS MONDAY 7 SEPT OR THURSDAY-FRIDAY NOV 26-27**

**FINAL EXAM TUESDAY 8 DECEMBER, 3:30 – 5:30 P.M.**

**( Course listed FIRST in the Timetable has precedence. )**

## 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	2.5 points each
<u>Lab Score</u>	<u>20 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. A program that does not compile and run cannot, by definition, solve any problems. However it may still be efficiently written using good programming style. Conversely, programs that happen to solve all problems may still be poorly written.
2. Graders may, at their discretion, give partial credit for any of the above categories. They are not, however, obligated to do so.

## Grading Pools

Because of the large number of students in this course, it will not be possible for the same person to grade all of the homework assignment submissions, even on a rotating basis. Therefore each TA will be responsible for grading the assignments of two of the lab sections, corresponding to the sections that they direct each week ( approximately 48 students each. ) In order to alleviate concerns over possible inconsistencies in grading between the different TAs, each student's scores will only be compared against the other students who were graded by the same TA. 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 readme 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 may be given a zero, at the grader's discretion.**
- **Graders are under no obligation to grade any program that does not compile.** However, they may choose to do so at their own discretion.
- The purpose of the readme file is to explain what the program does and how it does it, so that any reasonably competent programmer can easily understand it. It is in your best interest to make the grader's job as easy as possible by submitting a readme file that is both easy to read and which clarifies your code.
- 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.