

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

## Course Policies – Summer 2018

**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 logical thought patterns for implementing engineering solutions, including linear, branching ( conditional ), and repetitive execution, and to implement these concepts in Matlab and/or 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.
- Develop, write, test, and debug simple computer programs in C/C++ for solving engineering problems.

**Instructor:**

John Bell  
jbell@uic.edu  
<http://www.cs.uic.edu/~jbell>  
921 SEO, 312 413-9054  
Office Hours: May be 12-1:00 TTh, TBD Wed.  
See web for details  
Open Door policy during other times.

**Teaching Assistants:**

Shaika Chowdhury, schowd21@uic.edu  
See web site for updated 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:**

2249 SEL. Labs start the FIRST WEEK of classes.

**Required Textbook:**

The required textbook for this term is an online e-book from Zyante.com, with two types of embedded interactive content:

- “Participation” questions are largely multiple choice, with an infinite number of tries to get it right and answers provided. These should be completed as part of the assigned readings, generally before each class. Mastery of the participation questions will be checked at the beginning of each class, using i-Clickers. ( See below. )
- “Challenge” questions are open ended, do not have answers provided, and generally require a small amount of programming. These should be completed shortly after each topic is covered in class, generally by Fridays. Completion of the required challenge questions **by the assigned deadline** will be checked through the Zyante system.
  - **NO CREDIT** will be given for Zyante problems completed after the assigned deadline. However there is still *value* in completing the problems, as they may help you prepare for exams, or better, for what you need to know after leaving the course.
- See the course web site for details on how to register for the Zyante e-book, and for specific assignments and deadlines.

### Other Recommendations:

- Tutorials built in to Matlab. See “Getting Started”.
- Delores M. Etter and Jeanine A. Ingber, "Engineering Problem Solving with C++, Third Edition"
- Brian H. Hahn and Daniel T. Valentine, "Essential MATLAB for Engineers and Scientists", Fifth Edition, Elsevier, ISBN 978-0-12-394398-9
- Ray Lischner, “C++ in a Nutshell”, O’Reilly, 2003.
- Doug Brown and Gregory Satir, "C++ The Core Language", O'Reilly ( Nutshell Handbooks ), 1995.
- Brian W. Kernighan and Dennis M. Ritchie, “The C Programming Language”, 2<sup>nd</sup> Ed.
- <http://www.cplusplus.com>

### iClickers

This semester we will be using an interactive student response system called iClickers, which requires you to purchase an iClicker device from the bookstore or other source. You may purchase either version 1 or 2, but you must purchase a device – The applications that run on a phone or laptop are too slow to be usable in class. You may use the same device in more than one course, but a single device must be registered to a specific student, as follows:

Students must register the serial number on the back of their iClicker device in the Blackboard Learn course site for a particular course in order to get credit in their responses in that course:

1. Log in to the CS 109 Blackboard site.
2. Select iClicker Registration from the list of items on the menu.
3. Enter the serial number of your iClicker device in the iClicker Remote ID box.
4. Click Submit to confirm the registration.

See <http://acc.uic.edu/service/education-technology-support/clicker> for more information.

### Programming Assignment Grading Guidelines

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

Program compiles and runs ( using ACCC lab 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.
4. Optional enhancements may offer a chance to get back points lost elsewhere on an assignment, but may not raise scores above 100. It should be possible to earn 100 on assignments without optional enhancements, but only for **perfect** work.

## Overall Course Grading Policy:

Numerical scores will be based upon the following contributions:

(2) Midterm Exams ( 50 minutes ) * 10 points each =	20 points
Final Exam ( 2 hours )	20 points
(7) Programming assignments * 5 points each =	35 points
(3) Zyante challenge problem assignment sets <sup>1</sup> * 5 points each =	15 points
i-Clicker scores, mostly based on Zyante participation questions <sup>2</sup>	5 points
<u>Lab Score</u>	<u>5 points</u>
Total:	100, at end of term

- At any time during the semester your Percentage of Possible Points, PPP, can be calculated as:

$$PPP = \frac{\text{Points accumulated to date}}{\text{Possible points to date}} * 100\%$$

- The PPP will be posted to Blackboard, and will be updated as each assignment is completely graded. ( I.e. the PPP will only be updated after the entire class has been graded for the latest assignment. )
- **Mapping of PPP score to letter grades will not necessarily follow the traditional 90/80/70/60 grade breaks**, however some effort will be made to keep it close. Detailed feedback including an estimate of projected letter grades will be made after each of the midterm exams is graded and returned.
- Completion of the assigned Zyante challenge problems and iClicker results will be compiled into a combined scores three times during the semester, just prior to each exam<sup>1,2</sup>.
- Lab scores will be based on ***participation*** in lab, not merely attendance. It will be up to individual lab TAs discretion as to how to evaluate participation, however each student may miss up to two free lab period(s) without penalty.

## Minimum Requirements to Pass

Regardless of the overall numerical score, the following minimum requirements must be met in order to pass this course:

- Pass a majority of the exams, particularly the final exam. ( If you ace all your homeworks and fail the exams, you still fail the course. )
- Complete a majority of the programming assignments. ( You can't just show up on exam days. You have to complete your work too. )
- Participate in a majority of the laboratory sessions. ( Ditto. )

## Grading Pools

~~At the end of the semester separate grading curves ( mapping of PPP to letter grades ) will be used for each TA, to reflect the fact that not all TAs grade or teach identically.~~

---

<sup>1</sup> Completion of Zyante challenge problems will be checked at stated deadlines, approximately weekly. Three times per semester, the average of 5 lesson's worth of Zyante challenge problems will be compiled into a single score for that five-lesson period. The weekly scores may or may not be weighted to reflect varying number of problems each week.

<sup>2</sup> i-Clicker scores will be posted regularly, usually after each lecture. The scores will be compiled into a ( weighted ) average ( at least ) 3 times per semester, just prior to each exam. There will be one i-Clicker average score for the term.

### **Planned Exam Schedule:**

A detailed schedule of activities, including assigned readings, required Zyante activities, and assignment deadlines will appear on the web site, and will be developed as the semester proceeds. In addition the following exam schedule is planned as of this writing:

- Thursday 5 July, ( lecture 6 ), covering the 1<sup>st</sup> 5 lectures of material.
- Thursday 26 July, ( lecture 12 ), covering the 1<sup>st</sup> 10 lectures of material.
- Thursday 9 August, ( finals week ), covering the entire semester.

### **Exam Policy:**

- 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 original programs must be turned in using Blackboard.
- Each program must be accompanied by a user documentation file.
- Programs must compile and run properly on the ACCC lab computers to receive full credit. **Programs which run on other systems 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.
- All appeals for grading errors, no matter how justified, must be submitted within two weeks after graded assignments are returned. No grade appeals will be heard after that time.
- There are no provisions for making up missed lab days, other than the two free misses. On rare occasions students may attend an alternate lab session, with prior arrangement.

### **Late Days**

- All students are allowed 4 late days during the course of the semester: The first two late days are at no charge. The third costs a 10% late penalty. The fourth a 20% late penalty.
  - Late penalties are applied as a multiplier of 0.9 or 0.8. So a submission that would be worth 70 points normally would be worth 63 with a 10% penalty applied, or a 56 with a 20% penalty applied.
  - A late day is a 24-hour grace period, and is not divisible.
  - No more than one late day may be used on any given assignment.
  - Late days only apply to programming assignments, not the Zylante challenge problems.

## 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.
- **Clicking someone else's iClicker or signing an attendance sheet for someone who is not physically present are deliberate acts of dishonesty**, and will be reported to the Dean's office as ethical violations.

## CS 110 ( Does not apply during the summer )

Starting in Spring 2017, a special version of CS 109 is being offered, numbered as CS 110. Here are the differences between the two courses:

- ~~CS 110 students complete the first 5 weeks of CS 109, including the first midterm exam in the 6<sup>th</sup> week. These five weeks and this exam will cover Matlab only, with no C++.~~
- ~~CS 110 students will receive 1 hour of credit, instead of 3.~~
- ~~There are no adds or drops to/from CS 110 after the first week of classes.~~
- ~~The deadline to withdraw from CS 110 with a "W" is the end of the third week of classes.~~
- ~~CS 110 grades will be based on the same requirements as the rest of the class, through the first 5 weeks, as compared to the performance of the rest of the class over the same material.~~
- ~~CS 110 grades will be reported at the end of the term, along with all other grades.~~
- ~~CS 110 students are allowed one late day at no penalty, and no others.~~