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@cs.uic.edu  
http://www.cs.uic.edu/~jbell  
921 SEO, 413-9054, or ERF 2032 / 2072  
Office Hours: Initially TTh 12:00 - 1:00 and after class.  
See web for details  
Open Door policy during other times.

Teaching Assistant: Dibyayan Das, ddas6@uic.edu, SEO 805, MW 1:00-3:00  
See web site for updated office hours and locations

Corequisite: 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.

Textbooks:

Required:


Other Recommendations:

- http://www.cplusplus.com
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 optional - See grading policy later in this document.

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: **ILLI-19774-ZRAH-25** 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 rough schedule is planned, as of June 2015, and is subject to dynamic adjustment as necessary. A more detailed schedule, including reading assignments and links to online notes, will appear on the course web site.

<table>
<thead>
<tr>
<th>Section</th>
<th>Lectures</th>
<th>Topic</th>
<th>Notes</th>
</tr>
</thead>
</table>
| I       | 1 - 5    | Course Intro, Fundamental Development Tools - pseudo code, flow charts, simple samples, IDEs. Matlab fundamentals and functions | MIDTERM I 
THURSDAY JULY 9 
3:30-5:30 P.M. (see below) |
| II      | 6 - 10   | C/C++ - Simple programs through writing functions, including decision-making constructs and repetition. | MIDTERM II 
THURSDAY JULY 23 
3:30-5:30 P.M. (see below) |
THURSDAY AUG 6 
1:30 - 3:30 P.M. |

MIDTERM EXAM I, THURSDAY 9 JULY, 3:30 - 5:30 P.M.
MIDTERM EXAM II, THURSDAY 23 JULY, 3:30 - 5:30 P.M.
FINAL EXAM THURSDAY 6 AUGUST 1:30-3:30 P.M.

Notes:
- Midterm Exams will be 3:30 to 5:30 P.M., unless there are excessive conflicts, in which case they will be rescheduled for 6:00 to 8:00 P.M.
- The Final Exam is officially scheduled during class time, 1:30 to 3:10. We plan to continue to 3:30 unless there are unavoidable conflicts.

**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 either 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 actual performance on lab activities. Lab scores will be independent of homework and exams.
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
Lab Score 10 points
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.

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.

For those students who pass a majority of the exams and complete a majority of the homework, the following minimum grades are guaranteed:

An overall normalized score of at least: will earn at least:

90 A
80 B
70 C

In practice the actual grade breaks will probably fall somewhere below the numbers listed above, depending on the difficulty of the exams and overall class performance, so if you fall below those numbers, don't give up.

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 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.
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.
- 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.
  o 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.
  o A late day is a 24-hour grace period ( to 10:00 a.m. Wednesday ), and is not divisible.
  o No more than one late day may be used on any given assignment, since no submissions can be accepted after the solutions have been presented.
- 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.