

# COMP 116, Spring 2015 Introduction to Scientific Programming

## Class Hours:

	Tuesday/Thursday 11:00 AM-12:15 PM by Marc Niethammer
	It is recommended that you bring your laptop to all class sessions. It is essential that you bring your laptop to quizzes and exams!
Class Room	SN 014 (Sitterson Hall)



## Instructors:

### Marc Niethammer

Office	SN219 (Sitterson Hall, 2nd Floor)
Office Hours	Tu 12:15pm-1:15pm (right after class) or by appointment
Office Phone	919-590-6149
E-mail	<a href="mailto:mn@cs.unc.edu">mn@cs.unc.edu</a>

### TA : TBD

Office	
Office Hours	
E-mail	

### TA : TBD

Office	
Office Hours	
E-mail	

### TA : TBD

Office	
Office Hours	
E-mail	

### TA : TBD

Office	
Office Hours	
E-mail	

**Catalog Description:** **COMP 116** Introduction to Scientific Programming (3).

An introduction to programming for computationally oriented scientists. Fundamental programming skills, using MATLAB and another imperative programming language (such as C). Problem analysis and algorithm design, with examples drawn from simple numerical and discrete problems. *Students can only receive credit for one of COMP 110, **COMP 116**, or COMP 121.*

**Prerequisites:** Please contact us if you are concerned about whether you have the background required for this course.

*MATH 231:* We assume familiarity with univariate differential and integral calculus, and the ability to manually solve a system of simultaneous linear equations.

*Computer Literacy:* Basic proficiency with using a personal computer, using a mouse and keyboard, word processing, email, and finding information off the internet is assumed.

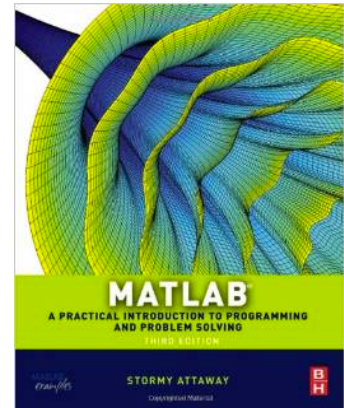
*Laptop:* Each student needs their own laptop computer for installing and using the MATLAB software during the course of the semester **as well as to take quizzes and exams.**

**Postrequisites:** **COMP 116** satisfies the *General Education, Quantitative Reasoning* requirement of 3 credit hours. For students interested in computer science as a major or minor, most upper level and graduate classes in computer science, assume that the student has learned basic programming by taking one of the following courses: COMP 110, **COMP 116**, or COMP 121.

**Disabilities:** If you have a physical or learning disability, please let us know about your special requirements. For more information, see the following website:  
<http://disabilityservices.unc.edu/>

**Text Book:** *MATLAB: A Practical Introduction to Programming and Problem Solving*, by Stormy Attaway, 3<sup>rd</sup> edition; Publisher: Elsevier, Inc, 2013.

**Web Site:** We use the Sakai web site, <http://sakai.unc.edu>, for access to handouts, email and discussion boards. You need your ONYEN to log into Sakai. (See <http://onyen.unc.edu> if you do not have an onyen. Forgotten passwords can be changed at the [IT Response center](#) in the basement of the House undergrad library, make sure to bring your student ID.)



**Piazza:** To allow for discussions of the material and to answer any questions you may have quickly we will make use of piazza (<http://www.piazza.com/>). Please sign up for an account.

**MATLAB software:** We will instructions on how to install Matlab 2014b available on Sakai. This can be done via an online registration. For those of you running into issues there will be a special session to make sure you have a working installation.

UNC has a site license that makes MATLAB available to all interested UNC students.

*Installation Requirements:*

- A Laptop to install MATLAB software on with a working DVD reader (internal or external)
- 700 MB - 2 GB free disk space on your laptop's hard drive.
- Internet access from your laptop
- A valid Onyen and password to login to the UNC MATLAB installation website
- A valid UNC email address <myname@mail.unc.edu> and password to register your academic MathWorks account.

**Course Objectives:** After taking this class, students will learn to do...

*Numerical Computation:* Use MATLAB for doing numerical computation: including arithmetic, algebra, calculus, working with Matrices, and solving systems of linear equations.

*Programming:* Create programs to solve scientific problems using MATLAB.

*Fundamentals:* Learn the fundamentals common to many programming languages (variables, data types, flow of control, modular design, etc.).

*Debugging:* Avoid and track down bugs using defensive programming techniques

*I/O:* Work with user input (or file input) and transform it into 2D & 3D graphics (or file output).

**Invitation:** Computers are powerful tools because of their flexibility. They can be programmed to deal with numerous tasks. Once people have learned how to write one program, they can, often with a little more programming, effectively use their area of expertise to best effect. For many modern

programming languages, including Java, C++, Python, Tcl/Tk, and Visual Basic, a key principle is to be able to write small fragments of code that interact with sophisticated tools that others have written.

This course introduces basics of programming that are common to many languages. Lectures and assignments draw examples from many scientific disciplines and focus on tasks like reading and analyzing data files and interacting with sophisticated analysis tools.

We focus on the programming language intrinsic to MATLAB, a common tool in mathematics and the sciences. MATLAB makes it easy to work with matrices, graphs, images, and sophisticated mathematical functions. This will allow us to do sophisticated computation (on images, protein molecules, terrain models, etc.) in realistic scenarios as we develop the fundamentals of programming.

Learning a programming language is like learning a spoken language -- *this has several implications*:

**Hands-on learning:** In our experience, texts and lectures alone are insufficient; one must spend time using a language to make it one's own. The weekly quizzes are to encourage practice in MATLAB.

**Journal:** Keep a record of things you learn in the lab, lecture, and recitation sections. Whether this is a formal lab notebook, a small diary, or jotted down notes on a word processor. Find a way that works for you to record your insights and ideas.

**Reading:** Students are strongly encouraged to keep current by reading and reviewing the assigned chapters, handouts, and in-class presentations. This is good preparation for the weekly quizzes, midterms, and the final. There is a lot to cover and learn. Cramming at the end of the semester is an unsuccessful strategy for this class.

**Elegance:** There is always more than one way to say something, but some ways will be more "elegant" than others. You will learn to recognize elegant expressions as you become more familiar with a language and to use them as you become more skilled.

**Seeking Help:** Language classes are cumulative; don't fall behind. Ask for assistance from your fellow students, the TA's, or from the Instructor, especially if you find yourself struggling.

**Attendance:** Attendance is recommended in order to stay on top of the material. Lectures and Exams will start promptly at the beginning of each class. Quizzes will start promptly 10 minutes before the end of each class. If you must be absent for an extended period of time due to reasons like an extended illness, please contact us to make arrangements to receive in-class handouts and assignments.

**Grading:** Your course grade is based on your cumulative performance on programming assignments, team project, quizzes, and exams as follows:

<i>Practice Quiz:</i>	2%		
<i>Quizzes:</i>	18%	<i>Assignments:</i>	18%
<i>Midterms:</i>	32%	<i>Final Exam:</i>	30%

You will be provided *instant* feedback during quizzes, assignments, midterms, and for the final exam if your solution is potentially correct. This will be done by provided test code. **Note that if you receive full score for the instantaneous feedback it does not mean that you will receive a full score during grading.** This is due to the fact that I will run the code you write against a different dataset, with

different values and different dimensions. **Therefore you will need to write code that is generally applicable rather than hardcoding solutions.** At the very least you should be able to write syntactically correct Matlab code. Matlab code that does not run (for quizzes, assignments, midterms, and exams) will receive 0 points: no partial credits will be assigned for code that cannot be executed.

**Practice Quiz (2%):** All quizzes, midterms, assignments, as well as the final exam will be taken online using Matlab. I.e., you will use a program called fetcher to download quizzes, midterms, assignments, and the final exam and to upload your solution. This of course requires you to have a working version of Matlab and an installation of fetcher (available for download on Sakai). To make sure that everybody has a working installation, your first graded task will be to download and submit a mock quiz. I will give you the solution for it and will demonstrate to you how the download/upload works. **This is extremely important, because you will have to do this over and over again throughout the course of the semester.** All quizzes, midterms, and the final exam are timed. You are responsible for your own timing: i.e., your time starts when you download a quiz/midterm/final exam and I will monitor when you upload your solution. You may submit as many solutions as you like (and I encourage you to do so in case you run into technical problems with your computer – e.g., run out of battery, Matlab crashes, etc.), **but only your last submitted solution will be counted! Late submissions will incur an automatic late penalty.**

**Quizzes (9 x 2% = 18%):** There will be short quizzes, which will be made available on Thursdays (and are due at midnight) for a total of ten quizzes taken. These quizzes will not be in class, but instead you will be able to take it any time during the day the quiz is due by downloading it. You will have 30 minutes to complete such a quiz. You will be responsible for your own timing. I.e., you will need to upload a solution latest after 30 minutes. Late submissions will incur a late penalty. You are free to submit as often as you like, but only your last submission will be graded.

No makeup is given for missed quizzes, but the lowest quiz score will be dropped at the end of the term for a total of nine graded quizzes. The quizzes occur regularly to encourage students to stay up to date on learning the material. *These quizzes will be taken and submitted online.* We will offer a practice tutorial on how to do this some time before the first quiz so that everybody will be familiar with it.

**Assignments (6 x 3% = 18%):** There are a total of 7 programming assignments. Typically every other week there will be an assignment to write a program. Doing the assignments is essential for learning how to program. Your lowest scoring assignment will be dropped for a total of 6 scored assignments. No makeup is given for missed assignments. **Assignments must be submitted before 5pm on their due date, which is typically on a Tuesday.** Assignments need to be turned in online. A matlab script to do so will be provided and you will receive instructions during class on how to use it. Late assignments will not be accepted and will receive a score of zero. Remember that programs communicate with people, not just the computer, so write readable code, use brief comments to clarify your intentions and to explain how you solved the assignment.

**Exams (2 mid-terms x 16% = 32%, Final = 30%):** There will be two mid-terms in class and one comprehensive final exam. You will be allowed to make use of Matlab during these exams as they will also be submitted online. If the campus is closed for any reason during a scheduled exam, then the exam will be given during the next scheduled class instead. Late submissions will incur a late penalty. You are free to submit as often as you like, but only your last submission will be graded.

**Computing the final letter grade:** Throughout the semester you will be able to see the scores you obtained on Sakai. To give you the opportunity to compute your current overall score a matlab script, *computeScore*, will be provided on Sakai. The eventual mapping from the score reported by *computeScore* (which uses the weightings discussed above) will be as given in the table below. There will be no rounding.

Score (in percent)	Letter grade
>=95	A
[90,95)	A-
[87,90)	B+
[83,87)	B
[80,83)	B-
[77,80)	C+
[73,77)	C
[70,73)	C-
[67,70)	D+
[63,67)	D
<63	F

**Academic Honesty:** Refer to UNC's honor code at <http://instrument.unc.edu/> for a discussion on Academic Honesty. If you are in doubt about any aspect of the honor code, please ask us.

**Collaboration:** Collaboration on assignments IS encouraged. However, what you hand in must be your own writing / typing. Good scholarship requires that all collaboration must be acknowledged. Thus, if you collaborate on the solution of a problem set, we expect that you list your collaborators in the comment section of your submitted code. Collaboration on in-class evaluations (quizzes, midterms, the final exam) is, of course, a violation of the Honor Code, see <http://www.cs.unc.edu/Admin/Courses/HonorCode.html>.

**Plagiarism:** Plagiarism will not be tolerated.

**Cheating:** Cheating in any form on assignments, quizzes, or exams will not be tolerated.

**Civility:** UNC places a priority on student learning. We value the inherent worth and dignity of every individual. We expect all to be civil to each other by

**Etiquette:** With the exception of laptops used to take notes or to run MATLAB, please keep all other electronic devices (IE cell-phones) turned off or silenced during class.

**Respecting Others:** Respect faculty, staff, students, guests, other people's private property, university property, policies, rules and regulations.

**Being Responsible:** Take responsibility for your choices and actions. Accept consequences for your inappropriate choices and actions.

**Being Professional:** Communicate in a professional and courteous manner in all forms, and at all times, whether verbal, non-verbal or written.

**NOTE!** The reading assignments below are NOT the only relevant sections in the textbook. You are encouraged to look for help on particular topics in various sections of the textbook. Also, the MathWorks has a comprehensive set of documentation on-line, as well as a vibrant on-line community. Please make use of these resources, within the bounds of your honor code.

**Course Schedule (subject to minor changes):**

DOW	Date	Lesson / Lab / Event	Readings / Handouts / Tests / Quizzes / Notes
Thursday	1/8	Intro to Course Intro to MATLAB	
Tuesday	1/13	Explanation of MATLAB Installation / Submission System  Assignment, Variables, and Expressions	<b>Practice Quiz #1:</b> <b>Assign #1 Handout:</b> Image Filtering <b>Handout:</b> Variable Naming <b>Reading:</b> 1.1-1.4
Thursday	1/15	Plotting & Publishing	<b>Reading:</b> 11.1, <b>Handout:</b> Scripts, Workspaces, Functions
Friday	1/16	Help Session for Matlab Installation Problems	<b>Time and Place TBD</b>
Tuesday	1/20	Intro. to Matrices & Vectors	<b>Reading:</b> 2.1-2.5 <b>Assign #2 Handout:</b> Water Statistics (plotting) <b>Practice Quiz #1 Due</b>
Thursday	1/22	Matrices & Indices	<b>Quiz #1:</b> Matlab as a Calculator & Vectors & Plotting
Tuesday	1/27	Matrices & Indices Intro to Debugging	<b>Assign #1 Due:</b> Image Filtering <b>Handout:</b> Coding Guidelines
Thursday	1/29	Matrices, Computations, & Linear Systems	<b>Reading:</b> 14.3 <b>Quiz #2:</b> Matrix Indexing
Tuesday	2/3	Least Squares & Linear Systems	<b>Assign #2 Due:</b> Water Statistics (plotting) <b>Assign #3 Handout:</b> Solving Linear Systems
Thursday	2/5	Conditional Logic	<b>Reading:</b> 1.6, 2.5, Chapter 4 <b>Quiz #3:</b> Matrices
Tuesday	2/10	Input / Output	<b>Reading:</b> Chapter 3
Thursday	2/12	Midterm #1 Review	<b>Quiz #4:</b> Conditional Logic
Tuesday	2/17	<b>1st Mid-term</b>	<b>1st Mid-term</b> <b>(material covered from 1/8 - 2/10 inclusive)</b> <b>Assign #4 Handout:</b> Conditional Logic
Thursday	2/19	Functions	<b>Reading:</b> 3.7, 6.1, relevant sections of Chapter 10 <b>Assign #3 Due:</b> Solving Linear Systems
Tuesday	2/24	Function Challenges	
Thursday	2/26	Function Challenges	<b>Quiz #5:</b> Functions
Tuesday	3/3	Loops (For & While)	<b>Reading:</b> Chapter 5 <b>Assign #4 Due:</b> Conditional Logic <b>Assign #5 Handout:</b> Terrain Modeling
Thursday	3/5	Common Idioms	<b>Quiz #6:</b> Functions
Tuesday	3/10	<b>Spring Break</b>	
Thursday	3/12	<b>Spring Break</b>	
Tuesday	3/17	Strings	<b>Reading:</b> Chapter 7

		Text Manipulation	
Thursday	3/19	File I/O	<b>Assign #5 Due:</b> Terrain Modelling <b>Quiz #7:</b> 'while' & 'for' logic <b>Reading:</b> Chapter 9
Tuesday	3/24	Debugging	<b>Reading:</b> 6.5 <b>Assign #6 Handout:</b> Protein Data Bank
Thursday	3/26	Cell Arrays & Structures	<b>Reading:</b> Chapter 8 <b>Quiz #8:</b> Strings & File I/O
Tuesday	3/31	Review for 2nd Midterm	
Thursday	4/2	<b>2nd Mid-term</b>	<b>(Material covered from 2/19 - 3/26 inclusive)</b>
Tuesday	4/7	Advanced Functions	<b>Reading:</b> Chapter 10 <b>Assign #6 Due:</b> Protein Data Bank <b>Assign #7 Handout:</b> Cell Structures and Arrays
Thursday	4/9	3D Plotting / Visualization	<b>Reading:</b> Chapter 11 <b>Quiz #9:</b> Cell Arrays & Structures
Tuesday	4/14	3D Plotting / Visualization	
Thursday	4/16	Intro to GUI Programming	<b>Quiz #10:</b> 3D Plotting / Visualization
Tuesday	4/21	Intro to GUI Programming	<b>Assign #7 Due:</b> Cell Structures and Arrays
Wed	4/23	<b>Review for Final Exam</b> 1-3, 4.3 - 4.4, 5 - 7, 10.4	<b>Last Day of class</b>
Tuesday	4/28	<b>Comprehensive Final Exam</b> <b>(12:00 PM -2:00 PM)</b>	<b>Comprehensive Final Exam</b> <b>(12:00 PM - 2:00 PM)</b>

**Exam Schedule from:** <http://registrar.unc.edu/academic-calendar/final-examination-schedule-fall/>  
final-examination-schedule-fall/