UN5390: Scientific Computing I

Fall 2016

Course and Instructor Information

Course #	: UN5390 (cross-listed as BE5390, EE5390 and MA5390)
Course name	: Scientific Computing I
Location	: Fisher Hall 330
Lecture days and time	: TR, 8:05 am – 9:20 am

Instructor	: Dr. Gowtham
Contact information	: g@mtu.edu | EERC B39 | (906) 487-4096
Office hours	: By appointment

Course Objectives

---

1. Exposure to Linux command line, shell scripting, revision control system, and several free and open source tools and utilities to aid in design and development of computational workflows
2. Acquire/Enhance good programming and communication etiquette with an emphasis on readability and clarity of written code
3. Translate science and engineering problems into computer programs in a language of student's choice
4. Learn compilation, debugging and profiling techniques
5. Understand and troubleshoot the (sources of) errors in such programs

Suggested References

There is no prescribed/required text book for this course. Listed below are useful references and are usually available from the university library. Much of the material deemed necessary by the instructor will be made available to you.

1. Numerical Recipes – The Art of Scientific Computing
   W. Press, S. Teukolsky, W. Vetterling, B. Flannery; 978-0-521-88068-8
2. The Art of Computer Programming (vol. I, II and III)
   D. Knuth; 978-0-201-89683-1, 978-0-201-89684-8, 978-0-201-89685-5
3. The Practice of Programming
   B. Kernighan, R. Pike; 978-0-201-61586-9
4. The Science of Debugging
   M. Telles, Y. Hsieh; 978-1-57610-917-5
5. Linux Command Line and Shell Scripting Bible
   R. Blum, C. Bresnahan; 978-1-118-00442-5
6. Language and/or domain-specific literature
   Check with your advisor(s), mentor(s) and/or friend(s)

Grading Scheme

```
Final grade = 50% Assignments + 30% Project + 20% Active Participation

A  (4.00): 93% and above        AB (3.50): 88% - 92.99%
B  (3.00): 82% - 87.99%         BC (2.50): 76% - 81.99%
C  (2.00): 70% - 75.99%         CD (1.50): 65% - 69.99%
D  (1.00): 60% - 64.99%         F  (0.00): 59.99% and below
```

Active participation (AP) is measured by how well you come prepared and your ability to lead the problem solving process in front of the class. There is no *curving*, i.e., the grade you get is the grade you earned. Anything less than B for the final grade could jeopardize your enrollment, funding and/or immigration status. Extra credit opportunities are exclusively for earning a reference letter for graduate school/internship/fellowship/job applications, and not for making up the poor performance in (prior) assignments.

General Guidelines

1. Show up on time. If the door is closed, stay out and do not bother coming in

2. Show up prepared, and in a presentable and professional manner. You might be randomly picked on any given day to lead a solution process

3. No drinking, eating, facebooking, gaming, instagramming, sleeping, snapchatting, texting, tweeting, etc. during class. Cell phones must be in silent mode as well

4. Course material and assignments will be distributed, and assignments and project work must be submitted via GitHub. You may turn in partially completed assignments, without the fear of penalty even if you believe the solution is incorrect, to show timely and continuous work. You may incorporate my suggestions, if any, into your assignment and turn it in for grading. Late submissions will not be accepted.

   There are no restrictions on programming languages. You can use any language, more than one if necessary, for this course. The choice of languages should be guided by their applicability to your research endeavors, and their potential for parallelization. You are responsible for learning the language including its features and caveats.

   It is expected that you will work on the assignment consistently from the day it's handed out. If you have any questions, concerns and/or need further information, you must get in touch with the instructor at least 48 hours before it's due. Use an method [in person (on/off campus), email (with `UN5390:` or `BE5390:` or `EE5390:` or `MA5390:` in the subject field), phone call, etc.] that works for you to do so.

   Assignments and project reports must be typeset in LaTeX using the provided template. Microsoft Word, Notepad, handwritten notes, etc. will not be accepted. There is no need to explicitly include the code in the *.tex* document.

   It is acceptable (and often highly encouraged) to seek help from your classmates and/or others but any such help must be clearly and appropriately cited in the assignment. If you helped someone, you must include that information as well. Absence of proper citation and/or inability to explain your work will mean that someone else authored it. This, in turn, means that you will get no credit for the problem and the incident will be treated as a case of academic dishonesty.

   Submissions must be thoroughly spell-checked for typographical and/or grammatical mistakes. If the submissions contain such mistakes or be otherwise deemed difficult to understand, you will be required to work with the *Michigan Tech Multiliteracies Center*

5. Academic policies: http://www.mtu.edu/deanofstudents/academic-policies/

Tentative Timeline

Week 01 – 02
Review of Prep Work and Training Camps
Rules of the Game
Git, GitHub and UN5390
Compliance, Integrity and Security
A Brief History of Computing
Computational Workflow
Programming Etiquette
Library Instruction (2% AP)
Assignment #02 (5%)

Week 03 – 04
Review of Assignment #02
Statistics
Numbers
Errors
Program Compilation
Debugging and Profiling Programs
Integrated Development Environment
Assignment #04 (10%)

Week 05 – 06
Review of Assignment #04
Numerical Results
Numerical Methods – Finding Roots
Numerical Methods – Differential Equations
Mock Conference/Business Cards (2% AP)
Assignment #06 (15%)

Week 07 – No Class
Instructor at *CASC*, Alexandria, VA
Watch Silicon Valley and Webinars

Week 08 – 09
Review of Assignment #06
Numerical Methods – Integration
Numerical Methods – Matrices
Assignment #09 (20%)

Week 10 – 11
Beyond Desktop Computing
Parallel Computing and Programming
OpenMP
OpenMP Examples
Term Project (Starting Week #10)

Week 12 – No Class
Instructor at *SC16*, Salt Lake City, UT
Attend Webinar(s)

Thanksgiving Break

Week 13 – 14
Review of Assignment #09
OpenMP Examples (Continued)
Hardware Accelerators
Dealing With and Visualizing Data
Research Marketing (AP 2%)
Semester Summary

Remaining active participation credit is reserved for attendance (12%), and leading the solution process to a problem in front of the class (2%). You will be chosen randomly.

A weekly meeting with your research advisor and a weekly status report of research project (worth 1% each) are required during weeks 10 – 14. You can use these individual status reports to write a comprehensive report, iff your advisor needs it.

Your research advisor needs to send a one line email by 5 pm on Friday of finals week indicating how much of 25% you have earned as part of the term project.