PHYS 6860: Modern Computational Methods in Physics

Room 312, Physics Building, Mon. and Wed. 4:00-5:15pm

LECTURER: Assoc. Prof. Joern Putschke
Office: 318, Physics Building
Email: joern.putschke@wayne.edu
Phone: 313-577-5407

OFFICE HOURS: (per discussion in first class)

Textbook(s)/Materials

The C++ Programming Language (3rd Edition), Bjarne Stroustrup (1997)
Online Tutorial: http://www.cplusplus.com/doc/tutorial/
The ROOT Framework Documentation: http://root.cern.ch/drupal/content/users-guide
The GNU Scientific Library (GSL): http://www.gnu.org/software/gsl/
Numerical Recipes (C version)

Course Content
Introduction to Linux/Unix command line/terminal and basic shell programming. Basic control structures, data types, dynamic memory, I/O and functions in C++. Object Orientated Programming (OO) in C++, classes, inheritance, polymorphism. Using scientific C++ libraries for example ROOT and/or GSL. Applying the computational skills to implement modern numerical/computational methods essential in all fields of physics.

Course Philosophy
This course will combine lectures, in-class exercises and projects to provide a hands-on learning environment. The course will be quite challenging and will require students to spend a significant time working on projects.

Learning Outcome
This course will teach students the basic computational knowledge required in a wide variety of physics research. The choice of C++ is motivated by the fact that it is the de-facto computing language in science as well as in industry, therefore teaching physics student additional skills useful in-and outside the academic environment. In addition once a modern programming language is mastered, it is easy to transfer this knowledge to any other programming language (if needed) as well as to (scripting) software frameworks like MATLAB and Mathematica. Furthermore the learned computational skills will be utilized to discuss and apply them on a wide variety of numerical/computational methods essential in physics and physics research.

Grading
Your grade in the course will be determined by your performance on the homework, the two hourly exams, and the final exam:

Homework	30%
Projects	30%
Final Project	40%

Grading Scale:
A/A-: 80%-100%; B+/B/B-: 70%-80%; C+/C/C-: 60%-70%; D+/D/D-: 50%-60%; E 0%-50%.

Academic dishonesty
All of the graded assignments are designed to measure your individual understanding of the material. Working together on the homework assignments is not considered cheating but blindly copying of someone else's homework is. Remember, you are only cheating yourself and it won't help you in the examinations!

Week	Content
1	Introduction; Why Linux? The Linux terminal; Basic shell script programming (I/0 and loops)
2	Advanced shell script programming Introduction to C++: Compiling and (simple) Debugging Introduction to Makefiles/environment variables
3	Basic structure of a program Data types, operators and basic I/0 I/O with files (focus on text files)
4	**Simple numerical integration/differentiation** Control structures Functions
5	Arrays and strings Pointers/references and dynamic memory allocation **Sorting Algorithms** (Advanced Debugging) Data structures
6	Introduction to Object Orientated Programming Classes in C++
7	(More complex Makefiles) Class Inheritance (Polymorphism)
8	**Linear Algebra: Vector/Matrices** **Basic discussion of Matrix inversion/unfolding**
9	**Solving ODE** Scientific C++ libraries: Introduction to GSL (short intro to Mathematica)
10	**Simple MC integration** Random number generation → Monte Carlo Simulations (MC) Scientific C++ libraries: Introduction to ROOT
11	**Graphing and Fitting (introduction to chi2 method)** Data Storage (I/O): Trees/Ntuples
12	**Hands on: Example of "Data Analysis"** Final Project (examples are chosen from High Energy/Nuclear/Astronomy and Condensed Matter applications)
13	Final Project
14	Final Project Presentations Special Lecture (Guest Speaker/TBD)