Course syllabus: PHY 504 Computational Methods in Physics and Astrophysics Spring 2019

An introduction to procedural and object-oriented programming in a high-level language such as C++ or modern Fortran with examples and assignments consisting of rudimentary algorithms for problems in physics and astronomy. Students will use the UNIX/Linux operating system to write programs and manage data, and the course will include an introduction to parallel computing and good programming practices such as version control and verification. The course will prepare students for courses in algorithms and methods that assume a knowledge of programming. *Prerequisite:* None. *3 credits. Letter graded (A, A-, B+, etc.)*

Instructor: Prof. Alan Calder
alan.calder@stonybrook.edu
ESS 438

Meeting: MWF 9:00 – 9:53 AM, Mathematics S235.

Office Hours Tues. 10:00–11:30 AM, Wed. 2:30–4:00 PM. Other times by appointment.

Texts: *Fortran 95/2003 for Scientists and Engineers* (3rd edition) by Chapman.
Absolute C++ (6th edition) by Savitch.

Evaluation: 2 hour exams, homework assignments, a project, and a final exam. The percentages of the grade for these are 20% for each hour exam, 30% for homework, 10% for the project, and 20% for the final.

Homework: Homework will be assigned most weeks and will be due the following week. Late homework will not be accepted without prior permission.

Exams: Two hour exams and one final exam. Missed exams may not be made up! With advance notice and/or careful documentation of extenuating circumstances, an exam may be excused or accommodations made.

Purpose: This course is designed to prepare graduate students with no prior experience for the realities of modern scientific computing. The desktop computer running Linux (or some form of Un*x operating system) has become ubiquitous in the fields of physics and astronomy for a variety of purposes: numerically solving problems that cannot be easily solved analytically, analyzing or acquiring data from experiments and observations, writing papers or reports, or presenting results on the WWW. This course will allow students to attain a level of scientific computing literacy needed to thrive in this field. The course will focus on developing the skills needed carry out core tasks on modern computers running Linux (or Un*x) operating systems. This course will cover the following core topics:

- Using Linux (or Un*x) computer systems running X-windows.
- Most of the Fortran programming language.
- A subset of the C++ programming language

The course will also briefly introduce

- Some elementary numerical methods
- An introduction to the LaTeX typesetting system and the gnuplot plotting software.
- An introduction to parallel computing.

What To Expect: This course will require you to carry out numerous programming or other computing tasks on the MATHLAB Linux machines located in S235 of the Math Tower. It is likely you will have to spend a substantial amount of time writing and debugging programs in this laboratory setting. It may be possible in some cases for you to carry out some assignments on other computers, however, the instructor for this course may not offer any formal support for such efforts. The bottom line is that you should plan to carry out your work on the

MATHLAB machines or other machines specified by the instructor. The instructor may require you to turn in your assignments electronically or in the form of hard copy. Lecture notes will be provided via Blackboard.

Americans with Disabilities Act: If you have a physical, psychological, medical or learning disability that may impact your course work, please contact Disability Support Services, ECC (Educational Communications Center) Building, room128, (631) 632-6748. They will determine with you what accommodations, if any, are necessary and appropriate. All information and documentation is confidential.

Academic Integrity: Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong. Faculty are required to report any suspected instances of academic dishonesty to the Academic Judiciary. Faculty in the Health Sciences Center (School of Health Technology & Management, Nursing, Social Welfare, Dental Medicine) and School of Medicine are required to follow their school-specific procedures. For more comprehensive information on academic integrity, including categories of academic dishonesty, please refer to the academic judiciary website at http://www.stonybrook.edu/uaa/academicjudiciary/

SPECIAL NOTE REGARDING PLAGIARISM AND DISHONESTY: All instances of plagiarized work or academic dishonesty will be brought before the Academic Judiciary Committee. All parties involved (both the copier and the person who produced the original work) will be held accountable for any instance of plagiarism or dishonesty. Students are responsible for protecting the security of your programming assignments by making sure that your directories are not world readable. If you are unsure how to secure your home directory see the instructor immediately.

Critical Incident Management: Stony Brook University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of Judicial Affairs any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or inhibits students' ability to learn. Faculty in the HSC Schools and the School of Medicine are required to follow their school-specific procedures.

Additional Class Policies

- Student Responsibilities: Students will be expected to abide by all University regulations, procedures, requirements, and deadlines as described in the Graduate Student Bulletin and other University descriptions.
- Attendance: Students are expected to regularly attend classes and participate in the classroom experience.
- Assignments: All work on class assignments is to be carried out independently unless otherwise stated for a project. Computer programs developed for homework assignments should be developed exclusively by each student. Late assignments will not be accepted.
- Computer Use: All use of University owned computers and networks must be in accordance with the University Information Technology Policy.
- Passwords: Students are responsible for maintaining their MATHLAB computer account password. Lost or forgotten passwords will under no circumstances be accepted as an excuse for turning in homework assignments. If one loses or forgets a password, the instructor is unable to reset it.
- Classroom Behavior and Conduct: Students are expected to conduct themselves in accordance with the policy and responsibilities described in the Graduate Student Bulletin. Also, please
  - Arrive for class promptly.
  - Avoid behavior that is disruptive to the classroom, especially the use of cell phones.
  - Avoid web surfing during class.
  - Be familiar with material presented in previous lectures.
- Reporting of Grades: Grades will be posted on Blackboard. The instructor will discuss grades during office hours but will not report or discuss grades via email.

Outline of Lectures Spring 2019

Note that the lecture topics are subject to change depending on progress of the class. Exam dates will not change.

class	date	topic
1	Jan. 28	Introduction and history architectures, operating systems, and programming. Shells.
2	Jan. 30	Accessing the MathLab machines, bits and bytes, "Hello, World!" program.
3	Feb. 1	Editors, Globbing/filename matching, the compiler, compiling and running programs.
4	Feb. 4	Data types, Fortran statements, arithmetic expressions. List directed I/O.
5	Feb. 6	Real, integer, and mixed-mode arithmetic, intrinsic functions.
6	Feb. 8	Relational and logical operators, character variables, concatenation, STDIN & STDOUT.
7	Feb. 11	File permissions, ASCII and binary files, Block IF constructs, ELSE and ELSEIF clauses.
8	Feb. 13	Gnu debugger, Top-down design, algorithms, flow-charts. Named and nested IF constructs.
9	Feb. 15	CASE constructs, Safe divides, Obsolete Fortran, Plotting data with Gnuplot.
10	Feb. 18	WHILE loops, EXIT and CYCLE statements, DO WHILE loops, examples of loop usage.
11	Feb. 20	Numerical integration I.
12	Feb. 22	Numerical integration II, Convergence testing.
13	Feb. 25	Root finding examples. Version control.
14	Feb. 27	Latex typesetting system.
15	Mar. 1	**Exam I**.
16	Mar. 4	Formatted I/O, OPEN and CLOSE statements, Format descriptors.
17	Mar. 6	Arrays I.
18	Mar. 8	Arrays II.
19	Mar. 11	Subroutines.
20	Mar. 13	Argument association, array arguments, assumed size arrays, a simple sorting algorithm.
21	Mar. 15	Introduction to parallel computing.
-	Mar. 18	
-	Mar. 20	Spring Break.
-	Mar. 22	
22	Mar. 25	Fortran modules and functions.
23	Mar. 27	Real types, precision, and makefiles.
24	Mar. 29	Programming paradigms, V&V.
25	Apr. 1	Introduction to C++ I.
26	Apr. 3	Introduction to C++ II.
27	Apr. 5	Integrating the equations of motion. Euler's method for ODE's.
28	Apr. 8	Runge-Kutta methods for ODE's.
29	Apr. 10	C++ relational and Boolean operators, precedence rules, conditional and loop structures.
30	Apr. 12	C++ functions, local and global variables.
31	Apr. 15	C++ argument-passing mechanism, C++ arrays, C++ arrays in functions.
32	Apr. 17	C++ multi-d arrays, C++ pointers, C++ dynamic arrays.
33	Apr. 19	File I/O in C++, function parameters..
34	Apr. 22	**Exam II**
35	Apr. 24	Structures in C++, derived types and modules in Fortran 90.
36	Apr. 26	Fortran pointers and linked lists.
37	Apr. 29	Type-bound Fortran procedures, classes in Fortran and C++.
38	May 1	Object-oriented programming, encapsulation.
39	May 3	Type extension Fortran types, C++ derived types, inheritance.
40	May 6	Overriding, polymorphism in Fortran and C++.
41	May 8	TBA
42	May 10	TBA
Final	TBA	**Final exam** (Comprehensive)