PHY 277: Computation for Physics and Astronomy Majors (3 credits)

Time & Location: Section 1 MWF (10:00-10:53) Section 2 MWF (9:00-9:53) S235 Math Tower (the Math SINC Site)

Instructor: Prof. F. Douglas Swesty
Office: Room 463 Earth & Space Sciences
Phone: 632-8055
Email: douglas.swesty@stonybrook.edu
Office Hours: TBD

Purpose:

This course is designed to prepare sophomore Physics and Astronomy majors 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 help you to attain a minimal level of scientific computing literacy that you need to function on a daily basis 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.

We will also touch briefly on the following topics:

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

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 and Teaching Assistants for this course will not offer any formal support for such efforts. Therefore 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, via web pages that you develop, or in the form of hard copy. The course T.A. will hold office hours in the MATHLAB in order to assist you with problems that you may encounter in carrying out your assignments. Lecture outlines will be provided via the course web site.

Recommended texts:

Note: These texts are **recommended but not required.** If you can purchase an earlier edition of either book for a lower price please do so. **Latest editions are not needed!**
Fortran 95/2003 for Scientists and Engineers, (or the Fortran 90/95 edition), Stephan J. Chapman
Absolute C++, (Any edition)., Walter Savitch

Course Grading & Attendance:

The grading for the course will be based on the completion of weekly homework assignments (30%) , unannounced quizzes (10%), a midterm exam (30%), and a final exam (30%). Each of these components

will count equally towards the final grade. **The course grades are curved.** The homework problems will consist mainly of computer programming assignments which are designed to emphasize the subject matter discussed in the preceding lectures. The programming assignments will be demanding and it will typically require several hours of your time per week to complete the assignment. If you fail to keep up with the weekly assignments you will have difficulty passing this course! Students will not be allowed to retake quizzes missed due to absence from class.

If an absence occurs causing you to miss an exam, quiz, or homework assignment due to a legitimate reason (illness, medical issue, death of a family member, jury duty, military service) please provide full documentation of the reason to the Office of the Dean of Students (222 Students Activities Center, 631-632-7320) and ask them to contact your instructors. If a quiz is missed due to an excused absence it will not be counted towards the quiz average.

Important University Policies:

Student Accessibility Support Center (SASC) Statement:
If you have a physical, psychological, medical or learning disability that may impact your course work, please contact the Student Accessibility Support Center (SASC), ECC (Educational Communications Center) Building, room 128, (631) 632-6748. They will determine with you what accommodations, if any, are necessary and appropriate. All information and documentation is confidential. Students who require assistance during emergency evacuation are encouraged to discuss their needs with their professors and the staff at the Student Accessibility Support Center (SASC). For procedures and information go to the following website: http://www.stonybrook.edu/ehs/fire/disabilities

Critical Incident Management Statement:

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.

Academic Integrity Statement:

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/commcms/academic_integrity/index.html

SPECIAL NOTE REGARDING PLAGIARISM AND DISHONESTY: All instances of suspected plagiarism or academic dishonesty will be brought before the Academic Judiciary Committee. All parties suspected (both the copier and the person who produced the original work) will be held accountable for any instance of plagarism or dishonesty. You 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.

Important Course Policies:

- **Student Responsibilities.** You will be expected to abide by by all University regulations, procedures, requirements, and deadlines as described in the *Undergraduate Student Bulletin.*

- **Attendance.** As per the University policy outlined in the *Undergraduate Student Bulletin*, students are expected to regularly attend all classes and to participate in the classroom experience.

- **Assignments.** All work on class assignments is to be carried out **completely** independently. **DO NOT ASK OTHER STUDENTS FOR HELP OR ASK THEM TO DEBUG YOUR CODE.** Ask the the instructor or the TA for help if you need assistance. There will be no collaborative work on

assignments at any time. Computer programs developed for this course should be developed exclusively by you alone. Late assignments will not be accepted.

- **Copying of Code.** Never, **EVER,** copy code from any source for use in your homework assignments unless an assignment explicity states that you can do so. This includes sources such as web sites, books, and others. Any instances of suspected copying of code for assignments will be referred to the Academic Judiciary Committee in accordance with University Policies.

- **Computer Use.** All use of University owned computers and networks must be in accordance with the University Information Technology Policy.

- **Classroom Behavior and Conduct**
  You are expected to conduct yourself in accordance with the minimal undergraduate student responsibilities described in the Undergraduate Student Bulletin including:
  - You are expected to 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.

Lecture, Homework, and Exam Schedule:

A tentative Lecture schedule is given below but is subject to change. Any changes to this schedule will be made to the version of this schedule posted on the course web site. The homework assignments will also be posted on the course web site. The mid-term examination date is posted on the course web site. The final examination will be held at the time scheduled by the registrar. It is the students responsibility to schedule classes so as to avoid final examination conflicts.

Note: You must take the exams during the time scheduled for your section. Failure to do so will mean your exam will not be graded and your exam will receive a score of zero

Date	Subject
Lecture 1	Bash shell; Basic Un*x commands; “Hello World!” program in Fortran.
Lecture 2	Using the compiler. The Un*x file system. Absolute & relative paths. A brief history of Fortran.
Lecture 3	Using SSH. Binary data representation. Various data types in Fortran.
Lecture 4	Shell variables. Fortran character set. Fortran statements. Structure of Fortran programs.
Lecture 5	Constants & variables. Declaration statements. Assignment statements. Parameters.
Lecture 6	List directed I/O. Debugging strategies. Arithmetic. Intrinsic Functions
Lecture 7	Shell aliases. Fortran operators. Character variables. Using STDIN & STDOUT.
Lecture 8	Directory listings & file permissions. IF constructs.
Lecture 9	Wildcards. Top-down design. Algorithms. Nested IF constructs.
Lect. 10	Obsolete constructs. Gnuplot
Lect. 11	While loops. Summation using loops
Lect. 12	The LaTeX typesetting system
Lect. 13	Nested loops. CYCLE & EXIT statements
Lect. 14	Numerical integration examples

Lect. 15	Root finding examples
Lect. 16	Plotting with Gnuplot. Basic formatted I/O concepts.
Lect 17.	Formated I/O. Format descriptors.
Lect. 18	Basic file I/O. OPEN and CLOSE statements. IOSTAT= clause in READ statements.
Lect. 19	Introduction to Arrays
Lect. 20	More on Arrays
Lect. 21	Multi-dimensional arrays. Allocatable arrays. The RESHAPE, ALLOCATE, and DEALLOCATE statements.
Lect. 22	**Midterm exam October 19**
Lect. 23	Subroutines.
Lect. 24	Argument association in subroutines. Array arguments in subroutines. Assumed size arrays. A simple sorting algorithm.
Lect. 25	A simple sorting algorithm. Separate compilation & linking. Modules.
Lect. 26	Modules. Module Procedures. Implicit and Explicit Interfaces.
Lect. 27	Function subprograms.
Lect. 28	Extended precision real variables and expressions. Makefiles
Lect. 29	Makefile Macros. Introduction to C++
Lect. 30	Introduction to C++ part II
Lect. 31	Integrating the equations of motion. Euler's method for ODEs.
Lect. 32	Runge-Kutta methods for ODEs.
Lect. 33	C++ Relational and Boolean operators. C++ Precedence rules.
Lect. 34	C++ conditional and loop structures.
Lect. 35	C++ functions. Pass-by-value mechanism.
Lect. 36	C++ argument-passing mechanism. C++ arrays. C++ arrays in functions.
Lect. 37	C++ multi-d arrays. C++ pointers. C++ dynamic arrays.
Lect. 38	More on C++ arrays. File I/O in C++.
Lect. 39	Derived types and structs.
Lect. 40	Type bound procedures in C++ and Fortran
Lect. 41	Encapsulation and Inheritance.
Lect. 42	Inheritance and polymorphism.
Date scheduled by Registrar	**Final Exam (Check Registrar's Final Examination Schedule at http://www.stonybrook.edu** **Note: It is the students responsibility to schedule classes so as to avoid final exam conflicts. No conflict exams will be given**