

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

Time & Location: MWF (9:35-10:30); S235 Math Tower (the Mathlab)

Instructor: Prof. F. Douglas Swesty

Office: Room 463 ESS

Phone: 632-8055

Email: phy277@mail.astro.sunysb.edu

Office Hours: MWF 10:30-11:30 or by appointment

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 95 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 HTML.
- 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. 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, via web pages that you develop, or in the form of hardcopy. 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.

Required texts:

FORTRAN 95/2003 for Scientists and Engineers, 3rd Ed., Stephan J. Chapman
Absolute C++, 3rd Ed., Walter Savitch

Course Grading:

The grading for the course will be based on the completion of weekly homework assignments, a midterm exam, and a final exam. Each of these components will count equally towards the final grade. 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!

Important University Policies:

If you have a physical, psychological, medical or learning disability that may impact your ability to carry out assigned course work, contact the staff in the [Disabled Student Services office \(DSS\)](#), 128 Educational Communications Center, 632-6748/9. DSS will review your concerns and determine with you what accommodations are necessary and DSS will advise me. All information and documentation of disability is confidential.

Stony Brook University expects students to maintain standards of personal integrity that are in harmony with the educational goals of the institution; to observe national, state, and local laws and University regulations; and 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, and/or inhibits students' ability to learn.

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. Any suspected instance of academic dishonesty will be reported to the Academic Judiciary. For more comprehensive information on academic integrity, including categories of academic dishonesty, please refer to the [academic judiciary website](#).

SPECIAL NOTE REGARDING PLAGIARISM AND DISHONESTY: All instances of plagiarised 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. 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.

Additional 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 independently. There will be no collaborative assignments. Computer programs developed for this course should be developed exclusively by you alone. **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**

You are responsible for maintaining your MATHLAB computer account password. Lost or forgotten passwords will under no circumstances be accepted as an excuse for turning in homework assignments. If you loose or forget your password your password the instructor is unable to reset it for you. Your only recourse for a lost or forgotten password is to appeal for help from the MATHLAB administrator at help@mathlab.sunysb.edu

- **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.

Date	Subject
Sept 5	Introduction and history of computer architectures, operating systems, and programming.
Sept 7	The Un*x file system. A brief history of FORTRAN. "Hello World" program.
Sept 10	Globbering/filename matching. The Compiler. Compiling & Running the "Hello World" program.
Sept 12	Binary data representation, data types, FORTRAN statements, structure of FORTRAN program.
Sept 17	Constants & Variables, declaration statements, assignment statements, parameters, list-directed I/O
Sept 19	Real arithmetic, Integer Arithmetic, Mixed-mode arithmetic, Intrinsic Functions
Sept 21	Relational operators, logical operators, character variables, concatenation operator, STDIN & STDOUT

Sept 24	Top-down design. Algorithms. Flow-charts. Block IF constructs. ELSE and ELSEIF clauses.
Sept 26	Logging in w/ ssh. Using scp to transfer files. File permissions.
Sept 28	Naming block IF constructs. Nesting block IF constructs. IF statements. CASE constructs.
Oct 1	While loops. The EXIT statement. DO WHILE loops. Examples of loop usage
Oct 3	Iterative DO loops. Summation using DO loops. Nested DO loops.
Oct 5	Naming DO loops. CYCLE statement. Numerical integration example.
Oct 8	Numerical integration examples.
Oct10	Root finding examples.
Oct 12	Plotting with Gnuplot. Basic formatted I/O concepts.
Oct 15	Formatted I/O. Format descriptors.
Oct 17	Basic file I/O. OPEN and CLOSE statements. IOSTAT= clause in READ statements.
Oct 19	Introduction to Arrays
Oct 22	More on Arrays
Oct 24	Multi-dimensional arrays. Allocatable arrays. The RESHAPE, ALLOCATE, and DEALLOCATE statements.
Oct 26	Subroutines.
Oct 29	Argument association in subroutines. Array arguments in subroutines. Assumed size arrays. A simple sorting algorithm.
Oct 31	A simple sorting algorithm. Separate compilation & linking. Modules.
Nov 2	Mid-term examination
Nov 5	Modules. Module Procedures. Implicit and Explicit Interfaces.
Nov 7	Function subprograms.
Nov 9	Extended precision real variables and expressions. Makefiles
Nov 12	Makefile Macros. Introduction to C++
Nov 14	Introduction to C++ part II
Nov 16	Integrating the equations of motion. Euler's method for ODEs.
Nov 19	Runge-Kutta methods for ODEs.
Nov 21	C++ Relational and Boolean operators. C++ Precedence rules.
Nov 26	C++ conditional and loop structures.
Nov 28	C++ functions. Pass-by-value mechanism.

Nov 30	C++ argument-passing mechanism. C++ arrays. C++ arrays in functions.
Dec 3	C++ multi-d arrays. C++ pointers. C++ dynamic arrays.
Dec 5	A brief introduction to LaTeX.
Dec 7	A bit more LaTeX. Creating a basic web page.
Dec 10	File I/O in C++.
Dec 12	Least-squares fitting.
Dec 14	Structures and Classes in C++. Derived Types and Modules in FORTRAN 95.
Dec 21	Final Exam. 8:00AM-10:30 AM