



---

## Course Essentials: Communication, Content, and Structure

### Communication

**Office:** HN 1090J

**Office Hours:** Tuesdays and Fridays, 9:45 - 10:45, or by appointment

**Email:** [Stewart.Weiss@hunter.cuny.edu](mailto:Stewart.Weiss@hunter.cuny.edu)

**Telephone:** (212) 772-5469 (my office)  
(212) 772-5213 (Department office)

### Resources

**Textbook:** Bruce Molay. *Understanding Unix/Linux Programming*. Prentice Hall, 2003. ISBN 0-13-008396-8.

**On-line Material:** This page and all documents related to this course are available for download on the CUNY Blackboard. In addition, I will be posting all material on my [website](#) in the [home page](#) for this class.

**Computing Facilities:** Registered students will be given user accounts on the UNIX hosts in the 1000G lab of the Computer Science Department, located on the tenth floor of Hunter North. This lab is open 24 hours a day, 7 days a week and access to it is limited to students enrolled in certain courses. Students will also be able to use a secure remote login service such as *ssh* to access these accounts. See the section entitled **System Access** below for instructions on how to obtain an *ssh* client for your home computer.

### Prerequisites

*If you have not had CSci. 340, you will not be permitted to take this class unless I feel that you are ready in spite of the lack of the prerequisite.*

### Course Content and Objectives

The principal objectives of this course are to teach you (1) how to write programs on and for a UNIX platform, (2) how to work efficiently within the UNIX environment, (3) how UNIX is designed and structured, and (4) why it is such a sensible operating system. It presents several different views of the *UNIX* operating system:

- For the application developer, it examines the most important parts of the UNIX operating system's application programming interface (API);
- For the ordinary user, it examines the command level view of UNIX; and



- For the computer scientist and system administrator, it exposes the internal structure of the UNIX operating system.

The course also contains a compact introduction to the *Perl* programming language.

I have taught this course in many different ways. Things you might have heard about previous versions of this course are not necessarily true of this instance of it. This version of the course covers many of the essential features of *Perl*, has little emphasis on shells and actual UNIX tools, and significant emphasis on system programming.

Over the years, many of the students who have taken this course and graduated have contacted me afterwards to tell me that it was this course that landed them their first jobs. I hope it will have this effect on you too.

## Programming and System Access

The first choice is to use the new 1000G lab that has been refurbished and equipped with workstations that run Linux. This lab is open “24/7” and has 24 workstations. The advantage of this is that you will be sitting in front of the monitor of the Linux host and will not be subject to potential disconnections that can take place when working remotely, nor the slowness of the network. The disadvantage is that you have to be in school to use it.

The other choice is to work *remotely*. The Computer Science Department makes a UNIX host, *eniac.geo.hunter.cuny.edu*, available to students who have access to the lab. You will be able to access this host from any computer that has *ssh* client software. If you download the *ssh* client software to your home machine, you will be able to login from home.

There are several versions of *ssh*. *OpenSSH* is an open source version developed for the OpenBSD project. *PuTTY ssh* is another free version for *Windows* operating systems, available at <http://www.chiark.greenend.org.uk/~sgtatham/putty/>. A third free version, from the original makers of the now commercial version is the *Windows* client *ssh*, which can be obtained at <ftp://ftp.ssh.com/pub/ssh/>. The latest client there is *SSHWinClient-3.2.9.exe*.

You can also install Cygwin on your *Windows* machine to create a UNIX environment within which you will have full access to the shells and commands of UNIX.

## Use of Blackboard in the Course

I rely on Blackboard for communicating to all students and distributing documents. You must check it daily for announcements. I enable the *Discussion Board* so that students can have a free exchange of ideas.

I am making effective use of the Discussion Board to answer questions. I would like everyone to use the following protocol:

1. Check whether the question you want to ask has been asked and answered in the Discussion Board.
2. If it has been answered, you are finished. If not, post the question in the Discussion Board and



3. Send an email message to me asking me to look at the question on the Discussion Board and answer it there.
4. I will send you an email message when I have answered it, so that you do not have to "poll" it waiting for an answer.

## Expectations, Tests, Assignments, and Grading

We will cover a lot of material. You are expected to do all of the specified reading, complete all assignments *on time*, and work independently. This is a seminar, not a regular lecture. As such, you are expected to be self-motivated and self-disciplined. If I want you to work in a group I will say so. Otherwise your work is to be yours alone. Class participation, project grades, and the final exam will be the basis for your final grade.

## Final Exam

There will be a Blackboard-based, multiple choice, final exam at the end of the semester, during the regularly scheduled final exam period, which is May 23rd from 11:30 to 13:30.

## About C and C++ In This Course

Most of the programming examples that I use and that appear in the textbook are written in C. Some students have a knee-jerk reaction when they hear this, saying, "but I don't know C." This is not completely true. The C++ language contains most of the C language. If you know C++, you know a great deal of C. There are minor differences that arise in the syntax of declarations (such as struct and function declarations), but the real problem is that most students never learn how to use the C standard I/O library. Most students learn stream I/O and never bother to learn the seemingly archaic functions of the C standard I/O library. In general, you ought to know C, if you want to call yourself a programmer. While you are free to write C++ code, you will be required to read C code.

## Academic Honesty Policy

Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The college is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures. In this class, I will enforce the University's Policy on Academic Integrity and bring any violations that I discover to the attention of the Dean of Students' Office.

## Make-up Policy, and Incomplete Grades

I do not give out incomplete (IN) grades except for those students who have been completing all work on time and who, for legitimate, documented medical or personal reasons, miss the final exam.