Essentials: Communication, Content, and Structure

Communications

Class Meetings:	Monday, Thursday 09:45 - 11:00
Office:	HN1090J
Office Hours:	Monday 12:00 - 14:00.
Email:	stewart.weiss@hunter.cuny.edu
Telephone:	(212) 772-5469

Resources

Textbook: Walter Savitch. *Absolute C++, Fifth Edition.* Pearson, 2012. ISBN-13: 978-0-13-283071-3; ISBN-10: 0-13-283071-X.

Computing Facilities: All registered students will be given user accounts on the Computer Science UNIX network if they do not already have one. These accounts provide access to all UNIX hosts in the network, including those in the 1000G lab on the tenth floor of Hunter North. This lab is available 24 hours a day, 7 days a week, to students enrolled in selected courses. The accounts also enable students to remotely login to the network using an *ssh client* by connecting to the gateway host `eniac.geo.hunter.cuny.edu`, and then *ssh*-ing to a lab host from eniac.

Course Website: Most course materials, including lecture notes, slides, assignments, syllabi, and other resources, including this document, are posted on my website, at `http://www.compsci.hunter.cuny.edu/~sweiss/course_materials/csci135/csci135_fall12.php`

My “demos” directory Programs that I have written to demonstrate various concepts, which I call *demo programs*, are not posted on the course website. Instead, they are posted on our file server in the directory
`/data/biocs/b/student.accounts/cs135_sw/demos/`.
All students registered in the class will have permission to access this directory. It can be reached from any computer on the department side of the firewall, such as the *Lab 1000G* computers, and remotely through `eniac`.

Publisher's Website The publisher of the textbook has many useful resources on their website, which is http://www.pearsonhighered.com/savitch/
If you have purchased a new copy of the textbook, then you will have access to the publisher's “companion website,” which contains video notes tied to the chapters of the book. Visit the URL
`http://wps.pearsoned.com/ecs_savitch_cpp_5/`
and register your student access code, after which you can login to watch these online videos.

Pre- and co-requisites

If you have not had CSci. 127, you will not be permitted to take this class unless there is reason to believe that you are ready in spite of the lack of the prerequisite. CSci 136 is a co-requisite for this class, which means that you must take it in the same semester in which you take this class. Although you do not have to take the section of CSci 136 taught by me, it is generally easier for you if you can.

Departmental Learning Goals

Material in this course supports or partially supports departmental learning goals: 2a, 2b, and 2c. These goals are written on the department's webpage,

`http://www.hunter.cuny.edu/csci/for-students/learning-goals-for-hunter-collegestudents.` .

Course Objectives and Overview

This course is an introduction to program development, problem solving, and to a lesser extent, the basic concepts of software engineering. Its emphasis is on the process of designing, implementing, and evaluating small-scale programs. Although it does cover parts of the C++ programming language, it is more than a C++ programming course. C++ is an extremely large and complex language with many features that interact in unexpected ways. One does not need to know even half of the language to use it well. Although the details of C++ must be mastered, they are nothing more than a distraction from the real goal, which is to learn how to analyze real world problems, develop and specify solutions to them, and design and implement their solutions. These skills will transcend the details of any specific language.

This is something like learning how to drive a car. You cannot learn to drive without sitting behind the wheel of a car. You need to know how to operate a particular car in order to drive, but the real objective is learning how to control your vehicle in all circumstances and to safely arrive at your destination, and this entails a deeper understanding of vehicles, traffic, and the road in general. There are principles involved. The same is true of learning to cook; you need to cook specific dishes in a specific kitchen to learn the principles, but it is not the specific recipes or tools that matter; it is an understanding of the principles of cooking.

Doing Well in This Course

In order to do well in this course you need to:

- do all of the assigned homework and assignments and hand them in on time;
- do the Self-Test Exercises in the chapters that we cover – the answers to all of them are at the end of each chapter; use these to test your understanding of the material; make sure that you attempt to do the exercises, not just look at the solutions!
- write code for some of the Programming Projects at the end of each chapter; we will work on some of them in the CSci 136 lab, but the more you program, the better you get at it;
- study the new material from one class to the next, not just before the exams;
- read the book's chapters *before* the class in which they will be discussed, and come to class with any questions about them;
- attend the classes;

- ask questions in class, on the discussion group, or after class when time permits.

Once again, the only way to learn programming is to spend many hours writing programs (and not only the programming assignments, but practice programs)! On average this class will require about one hundred twenty hours of your time outside of the lecture itself over the course of the semester.

Assignments, Exams, and Grading

Your grade will be based on exams (60%), programming projects (30%), and homework assignments (10%).

Exams

There will be two midterms and one final exam. The midterm exams are worth 18% each, and the final is worth 24%. In the midterm exams you may be asked to write programs or pieces of programs to solve problems. You will also be asked questions that test your understanding of concepts taught in class. If you understand the programming language features and the more general concepts that arise in the course of the semester, and you can think well under the pressure of an exam, you will do well.

All exams are closed-book, closed-note exams. The first midterm covers material from the beginning of the semester up to the week of the exam; the second covers the material from that week through the week of the second exam. The final covers the material from after the second midterm through the last day of class. Although each exam primarily covers new material, it also includes some cumulative material.

Programming Projects

You will be judged on your ability to solve problems, write programs, and work independently. No one can learn to program without writing programs. There will be *three* programming projects during the semester. They count 30% in total towards your grade, 10% each. Their dues dates (which are subject to minor changes) are in the table below.

Homework Assignments

There will be occasional homework assigned from the chapters of the textbook. In total the assignments are worth 10% of the course grade.

Exam and Project Schedule (revised)

Exam/Project	**Percent of Grade**	**Date**
Project 1	10	10/4
Exam 1	18	10/11
Project 2	10	**11/15**
Exam 2	18	**11/19**
Project 3	10	12/03
Exam 3	24	12/20 11:30 - 13:30

Lateness, Make-up Policy, and Incomplete Grades

All exams must be taken on time. Failure to take an exam counts as a zero grade on that exam. The only exception to these rules is for students who have a legitimate medical or personal emergency that prevents his or her timely completion of a project or sitting for an exam. You must present documentation to support a claim of this nature as soon as possible. Make-up exams will only be given to those students who legitimately missed the exam.

Late programming assignments lose 20% each day that they are late (i.e. a perfect assignment is worth only 60 points out of possible 100 points two days after the due date and zero points five days after the due date).

I will not accept any late homework assignments.

I will not give an incomplete (IN) grade to anyone other than those students who have been completing all work on time and who, for legitimate, documented medical or personal reasons, miss the final exam. There are no exceptions to this rule. For example, I will not give an IN grade to someone who has fallen behind on the projects and does not hand the last project in on time. You should acquaint yourself with the current Hunter College policy on grades in general, and in particular, on IN grades, which is found on this webpage: `http://catalog.hunter.cuny.edu/content.php?catoid=15&navoid=1433`.

Class Calendar

The last day to drop a class without a "W" is September 14. The last day to withdraw is October 9. There are no classes on September 3 and 17, October 8, and November 22. On Wednesday, October 10, classes follow a Monday schedule so we will have class on that day. The last day of class is Monday, December 10.

Programming and System Access

All students enrolled in the class are given accounts on the Computer Science Department's network. This entitles you to around-the-clock access to the 1000G lab, which is equipped with 28 Linux workstations. This lab is normally open "24/7". The account also enables you to work work from home or another remote computer by connecting to any of the lab machines remotely. The details are described below.

The advantage of working in the lab, as opposed to working remotely, is that you will be sitting at the console of a Linux host and will not be subject to potential disconnections that can take place when working remotely. You will also be much less affected by network problems than if you connect remotely from outside of Hunter. The disadvantage is that you have to be in school to do this.

When you are in the lab there are a few important rules that must be followed:

- Never power down a machine for any reason.
- Never leave a machine without logging out.
- Never use lockscreen to lock the screen in your login.

There are several other rules regarding lab use; they are posted there. Please take the time to read them and then follow them.

The Computer Science Department makes a UNIX host, named

`eniac.geo.hunter.cuny.edu`,

available to students who have accounts on the network. You will be able to access this host from any computer that has *ssh* client software. Once you login to `eniac`, you are requested to login from `eniac` to one of the machines in the 1000G lab, which are named `cslab1`, `cslab2`, `cslab3`, and so on, up to `cslab28`. You cannot *ssh* directly to those machines from outside of Hunter College for security reasons.

Many computers come with a version of *ssh* already installed. If yours does not, you can get one for free. There are several free versions of *ssh*. *OpenSSH* is an open source version developed for the *OpenBSD* project. *PuTTY ssh* is a free version for the Windows operating systems, available at

`http://www.chiark.greenend.org.uk/~sgtatham/putty/.`

Macintosh computers come with a command-line *ssh* client.

Course Materials, the Web, Blackboard, and the CSci Network

All lecture notes will be posted on the course's home webpage (whose URL is above), which does not require special privileges to access. The only thing for which I use Blackboard is for posting of grades, which will be posted in the grade center there. For the purpose of discussions and course-related questions, the class has a Google group with the following properties:

Name:	`hc_csci135-6_fall12`
Home page:	`http://groups.google.com/group/hc_csci135-6_fall12`
Email address:	`hc_csci135-6_fall12@googlegroups.com`

You will receive an invitation to join this group to your Hunter College email address. You should accept this invitation. Your Hunter email address can be used for reading and sending messages to the group, but unless you have a Google email address, you will not be able to access the group's home page to read old messages. If you do not have a Google email address, I suggest that you obtain one. If you do, you can request to join the group with that address. In fact, you can request to join the group with any email address you choose, and I will accept the request.

The Google group will be the means by which to ask and answer questions related to the course. You only need to subscribe once, even if you are also registered in my section of CSci 136.

I require that you use the following protocol if you have a question:

1. Check whether the question you want to ask has been posted and answered in the Google group.
2. If it has been answered, you are finished. If not, send the question to the Google group.
3. Anyone in the group can answer the question. If no one else answers the question in a timely manner, I will post an answer to it.

I will ignore any non-personal questions sent to my Hunter email address. Personal questions (such as a questions about a grade or a missed class or alternative times to meet with me) should be sent via private email to my Hunter email address, not to the Google group.

Academic Honesty

You are free to talk with others about the assignments, you can share ideas about program logic, about how to accomplish various tasks, about correct syntax and semantics, about what the input looks like or what the output should look like, and so on. You are not allowed to represent someone else's work as your own, no matter how little it may be. In other words, you cannot take someone else's program or part of it and say it is yours or put your name on it. Calling someone else's work your own is *plagiarism*. If you assist someone else to commit plagiarism by knowingly giving them your work so that they can attribute it to themselves, then you have participated in an act of academic dishonesty.

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.

Course Content

Following is a list of the topics from the textbook that will be covered. Some of these were covered in CSci 127 and will be reviewed quickly. The order listed here is not the actual order in which we will cover them. The course reading and exam schedule, poste don the website, has the exact order and schedule on a class by class basis.

Chapter 1: C++ Basics

- Introduction to C++
- Variables, Expressions, and Assignment Statements
- Console Input/Output
- Program Style
- Libraries and Namespaces

Chapter 2: Flow of Control

- Boolean Expressions
- Branching Mechanisms
- Loops

Chapter 3: Function Basics

- Predefined Functions
- Programmer-Defined Functions
- Scope Rules

Chapter 4: Parameters and Overloading

- Parameters
- Overloading and Default Arguments
- Testing and Debugging Functions

Chapter 5: Arrays

- Introduction to Arrays
- Arrays in Functions
- Programming with Arrays
- Multidimensional Arrays

Chapter 6: Structures and Classes

- Structures
- Classes

Chapter 7: Constructors and Other Tools

- Constructors
- More Tools
- Vectors–A Preview of the Standard Template Library

Chapter 8: Operator Overloading, Friends, and References

- Friend Functions

Chapter 9: Strings

- An Array Type for Strings
- Character Manipulation Tools
- The Standard Class string

Chapter 10: Pointers and Dynamic Arrays

- Pointers
- Dynamic Arrays
- Classes, Pointers, and Dynamic Arrays

Chapter 11: Separate Compilation and Namespaces

- Separate Compilation

Chapter 12: Streams and File I/O

- I/O Streams
- Tools for Stream I/O