

# CSCI 13500§01 - Software Analysis & Design 1

## Spring 2017

**Meeting Time:** Tuesday and Thursday, 5:35-6:50 PM, N-1036

**Instructors:**

Eric Schweitzer

eric.schweitzer@hunter.cuny.edu

212-772-4349

N-1000E

Wednesday and Friday, 3:30 to 5:00

or by appointment

Raffi Khatchadourian

raffi.khatchadourian@hunter.cuny.edu

212-772-5213

N-1000C

Tuesday and Thursday, 4:30 to 5:30

or by appointment

Send plain (ASCII) text email, from your “myhunter” account. HTML, MS-Word docs and email from other sources are likely to be ignored.

**Pre-/Co-requisites:** The prerequisite is CSCI 127 or instructor’s permission. At the very least, you should have written, compiled, and run a >1 page program containing iteration (while/for) and selection (if) statements.

CSCI 136 is a co-requisite, and you will not receive a grade in 135 if you don’t take 136. If you have already passed 136, you **must** contact the instructors before the 2nd class, so that you can be added to a bb section and your work can be graded.

**Questions:** Ask during class! Don’t let confusion stop understanding. Questions may also be asked on Piazza, in office hours, or of your 13600 instructor (especially for those needing hands-on assistance).

**Bulletin Board:** You should check the Blackboard (<http://bb.hunter.cuny.edu>) sites for both 135 and 136 regularly, since all class material will be posted there. Please make sure you have configured bb to use your preferred email address (your Hunter email address, by default), since you are responsible for any email the instructors might send there.

**Piazza:** This term we will be using Piazza for class discussion. The system is designed to get you help quickly and efficiently from classmates, the 13600 instructors, and ourselves. Rather than emailing questions to the teaching staff, we encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email [team@piazza.com](mailto:team@piazza.com).

Find our class page at: <https://piazza.com/hunter.cuny/spring2017/csci13500/home>

**Text:** Walter Savitch, Absolute C++, 6th Edition, ISBN 0-13-397078-7

You are responsible for all the material on the reading list whether or not covered in lectures.

**Grading:**

25% Programming Projects (3-5) This may not seem like a lot, but beware that many test questions will be based directly on these programs.

75% Tests (3, equally weighted). Tests 1 and 2 will be announced at least a week before they take place in class, on Piazza and on BlackBoard. Test 3 will be according to Hunter’s official final exam schedule for this class. This is currently scheduled for Tuesday, May 23, 5:20-7:20, but that is subject to change by the College.

**Late Policy:** Late penalties for programming projects are: 5% within 2 hours of the due date/time, 10% within 24 hours, and an additional 10% for each day after that. No homework is accepted after 1 week.

Make-up tests are not given under any circumstances. If there are extraordinary circumstances why you are unable to take a test, you must provide documentation *prior* to the test to avoid getting a zero.

**Software:** The standard Linux/Unix/MacOS C++ compiler is g++. There are several free Linux OSs which you can install for dual-booting with Windows, including <http://www.ubuntu.com>. If you want a Linux-style environment for windows that doesn't require installing Linux, <http://cygwin.com> is an alternative. You will be able to ssh into "CSnet" to compile and run your programs in our Linux environment. The G-Lab machines are the arbiter of "compile and run".

**Course Goals:** This course is:

- An introduction to software development, using the C++ programming language. Software development is a skill involving understanding the problem being modeled, as well as expressing a solution using a programming language. Thus, it has both conceptual and technical components. The successful student will be able to clearly and logically transform a problem, while being comfortable with C++ to express the transformed problem.
- A preparation for further courses in computer science. This course comprises the ABCs of computer science, and you're not allowed to forget it anymore than you can forget the alphabet after kindergarten. Students who expect to take more advanced courses in computer science need to go beyond "understanding" the material presented in this course – they need to master it.
- Time consuming; very, very time consuming. Any programming course takes up a lot of a student's time. In addition to the time spent in class, most students will need to spend between 10 and 20 hours a week at a computer. That makes for a total of 15-25 hours a week – that's 15-25 hours a week you must dedicate JUST FOR THIS COURSE, no kidding!

This course is NOT:

- An introduction to computers in general. we will not cover: Linux, networks, databases, etc. This course teaches a specialized skill – programming – and only programming. You should already possess basic computer skills such as compiling simple computer programs, editing files, manipulating files, etc.
- An overview of the C++ language. C++ is a huge language with a lot of highly technical details. We will cover the fundamentals of C++, but the focus is on designing algorithms and solving problems.
- A good idea to take if you are working full time and taking a full course load, or, for any other reason(s) you don't have a lot of free time to devote to CSCI 135. Although much of the material isn't especially difficult, it usually requires many, many hours to master. It is difficult to understand software development concepts without sitting in front of a computer many hours a week actually writing and debugging programs. Be honest with yourself. Make sure this course is for you, – now – at this point in your academic life. If you would like to discuss the time requirements further please feel free to come talk with the instructors.

This course provides a 'first step' towards the following department's **learning goals**:

- Have a deep practical knowledge of one widely used programming language
- Be experienced in working in at least two widely used operating system environments
- Be able to apply principles of design and analysis in creating substantial programs and have experience working in teams on projects of moderately realistic scope.

## How to Learn:

- From the beginning, students will be expected to work independently outside of the lectures. There will be very little “hand holding” in the course – you are expected to find your way around your computer on your own. For example, the way each of you will save your work, compile/debug an assignment, etc. will vary. These techniques will not be covered in class. Get started NOW (especially if you are going to install your own compiler and/or OS software). The first programming project is due soon.
- There will be many obstacles to overcome, both in absorbing the many examples of C++ programs and in doing the assignments. Attacking the obstacles head on, outside of class time, in front of a computer, is the key to success. Keep trying, if your program is not working, try again. Still not working? Try again. And again. And again. When it comes to programming, the learning is in the doing. There is no substitute for spending many hours in front of a computer – trying and failing, trying and failing, trying and failing – until you finally get your program up and running correctly. Every time you fail, you actually learn quite a bit, and to pass the course, you will repeat this trying-failing cycle many times, every week of every month during the entire semester. There are virtually no “slow points” during the semester.
- There will be approximately four programming projects. By far, the main cause for an unsatisfactory final grade is falling behind on the assignments. Exams are largely based on the programming assignments. If you don’t do the assignments, on time, you will almost certainly not pass the tests.

**Other:** We ask that all cell phones, pagers, etc. be silent in class. Any electronics in use should be used for class related activities. Violations of these “requests” will result in the loss of 2 points from you final average (per occurrence).

All course material (including lectures, solutions, etc.) is owned by the instructor and protected by **copyright**. You may use the material for yourself, but any other use (including posting on websites, whether free or not) is illegal without our express written permission.

We take **academic honesty** very seriously, and any violation results in an automatic F for the course along with sanctions in accordance with Hunter College procedure.

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 compliance with the **American Disability Act** of 1990 (ADA) and with Section 504 of the Rehabilitation Act of 1973, Hunter College is committed to ensuring educational parity and accommodations for all students with documented disabilities and/or medical conditions. It is recommended that all students with documented disabilities (Emotional, Medical, Physical and/ or Learning) consult the Office of Accessibility located in Room E1124 to secure necessary academic accommodations. For further information and assistance please call (212-772-4857)/TTY (212-650-3230).

This syllabus is subject to change should the need arise.