

CS 102 Syllabus

2 December 2012

Welcome to CS 102, an introduction to problem solving, algorithmic design, and implementation using the C++ programming language. Topics include fundamental data types and associated array types, I/O processing, conditional and loop constructs, use and implementation of functions. A brief overview of structures is given. Throughout the course, good programming styles and sound program construction are emphasized.

Monday, Wednesday 11am–12:50pm in LLC 207

Three credits, prerequisite: CS 101

Contact information

Instructor: Prof. Christopher League, Ph.D.

Email: christopher.league@liu.edu – please include course number in subject.

Messaging: chryslleague (AIM), league@contrapunctus.net (MSN), chrisleague (Skype), cleague@gmail.com (GTalk/XMPP)

Phone: +1 718 488 1274 (office), +1 646 450 6278 (Google voice)

Office hours: Monday 1:30–2:20; Wednesday 1:30–2:20 and 5:00–5:50

Office location: LLC 206

Resources

Software: VirtualBox 4.x, setup explained in first assignment.

Web sites: <https://liucs.net/cs102f12/>

Text: *C++ Without Fear* by Brian Overland (second edition, ISBN 978-0-13-267326-6)

Library: Campus library resources tailored for computer science are available at <http://www2.brooklyn.liu.edu/library/wlp/LibPortal-CS-BC.htm>

Requirements

Your grade will be computed based on assignments, exams, quizzes, and participation. There are a total of 1,000 points available, broken down as follows:

- There will be **12 assignments** during the semester. Assignments are worth **60 points each**, but I will **drop the lowest two scores** so only 10 will count, for a total of **600 points**.

- There are **8 quizzes** scheduled throughout the semester, to make sure you are following along and reviewing your notes after each meeting. Quizzes are worth **30 points each**, but I will **drop the lowest two scores** so only 6 will count, for a total of **180 points**.
- There will be a midterm and final exam, worth **100 points each** for a total of **200 points**.
- The remaining **20 points** are for your attendance and participation in class.

On the 1,000-point scale, you can expect the following letter grades:

	≥ 870: B+	≥ 770: C+	≥ 670: D+
≥ 930: A	≥ 830: B	≥ 730: C	≥ 600: D
≥ 900: A-	≥ 800: B-	≥ 700: C-	else: F

In the end, I may choose to adjust the scale slightly to compensate for assignments or questions that turned out to be trickier than I intended. Such adjustments would never *lower* your grade from what is designated in the above table; if you get 930 points, you are guaranteed an A.

Policies

No late assignments will be accepted, because we will discuss and evaluate your work promptly after the deadline. This helps to ensure that everyone receives timely feedback, and that you can learn from mistakes while they are still fresh in your mind.

There will be no extra credit. Students usually ask for extra credit late in the semester after they have already messed up their original opportunities. Be sure to start your work early, so that we can detect and solve any problems before they can affect your grade.

Plagiarism is the use or presentation of ideas, words, or work that is not one's own and that is not common knowledge, without granting credit to the originator. Plagiarism is a practice that is not only unacceptable, but which is to be condemned in the strongest terms possible on the basis of moral, educational and legal grounds. Under University policy, plagiarism may be punishable by a range of penalties from a failing grade in the assignment or course to dismissal from the School of Business, Public Administration and Information Sciences. All students are required to read the handbook on avoiding plagiarism by visiting <http://bit.ly/1VShWN>

Cheating includes, but is not limited to the following: falsification of statements or data; listing sources that have not been used; having another individual write your paper or do your assignments; writing a paper or creating work for another student to use without proper attribution; purchase of paper or research work for one's submission as his/her own work; using written, verbal, or electronic or other sources of aid during an examination (except when expressly permitted by the instructor, depending on the nature of the examination) or knowingly providing such assistance to aid other students.

Showing up on time to class every week is extremely important. If you must be absent or more than 5 minutes late, please try to notify me in advance. I will be keeping track

of whether you are in class, and when you arrive. A few missed classes will not count against you, but habitual absence will significantly hurt your grade. Additionally, there will be no make-up quizzes. I do not distinguish between ‘excused’ and ‘unexcused’ absence. Unless you miss an *exam* due to a severe medical emergency, I don’t want to see a doctor’s note. If you do miss an exam, the make-up exam will be different – and probably *not* easier.

Long Island University seeks to provide reasonable accommodations for all qualified persons with disabilities. This University will adhere to all applicable federal, state and local laws, regulations and guidelines with respect to providing reasonable accommodations as required to afford equal educational opportunity. It is the student’s responsibility to register with Special Education Services (SES) as early as possible and to provide faculty members with the formal communication from SES for suitable accommodations. All accommodations must be approved through SES. Contact Information: 718 488 1221 or 718 488 1044.

Time commitment

This is a lab course, for which you will have to spend a significant amount of time both inside and outside of class to succeed. In addition to spending about **1–2 hours** preparing (reading, reviewing, practicing) for each hour of class time, your work on the assignments is a crucial part of the learning experience. Some time will be set aside in class for supervised work on the assignments, but it will not be sufficient.

The productivity of computer programmers varies widely, depending on the project and skill level, by a factor of ten or more. (In other words, the most productive programmer can accomplish the same task in one-tenth the time taken by the least productive programmer.) This factor comes from studies of professional programmers; for beginners, the effect is probably multiplied even further.

For this reason, I am reluctant to estimate the number of hours a ‘typical’ student will need to spend on each assignment. However, the state of New York requires it, so here we go. On average, expect to spend **5 hours per assignment** (keeping in mind that earlier assignments will require less time than later ones), or a total of **60 hours per semester**. You may find you need less time, or you may find you need spend substantially more time, in order to achieve the educational goal. So please don’t get discouraged if you find yourself working even more than this. With practice, you will get there. Nothing worth doing is easy.

Goals and objectives

Upon completion of the course, students should be able to...

- demonstrate proficiency in basic algorithms and data structures (1.1, introductory level).
- understand the mathematical and logical foundations of computing (1.2, introductory level).

- understand the role of programming languages in software architecture (2.1, introductory level).
- use tools such as a compiler, editor, and development environment (2.2, introductory level).
- work with simple data models in a programming language (3.2, introductory level).
- exhibit awareness of professional organizations and technical opportunities (5.1, introductory level).
- productively attend seminars and workshops outside of class work (5.2, practicing level).

Schedule

Wed 5 Sep **Meeting 1** at 11 am: Introduction — languages, compilers, and other tools.

Mon 10 Sep **Meeting 2** at 11 am: Building a C++ program, output, and includes. *Read p. 1–18.* **Assignment 1** due at 1 am.

Wed 12 Sep **Meeting 3** at 11 am: Types, variables, and operators. *Read p. 19–25.* **Quiz 1.**

Mon 17 Sep **Meeting 4** at 11 am: Exercises. *Read p. 26–31.* **Assignment 2** due at 1 am.

Wed 19 Sep **Meeting 5** at 11 am: Decisions with if/else. *Read p. 33–42.*

Mon 24 Sep **Meeting 6** at 11 am: Nested if/else chains, exercises. **Assignment 3** due at 1 am.

Wed 26 Sep **Meeting 7** at 11 am: Introduction to loops, increment/decrement operator. *Read p. 43–52.* **Quiz 2.**

Mon 1 Oct **Meeting 8** at 11 am: Logical operators, short-circuit evaluation, exercises. *Read p. 53–65.* **Assignment 4** due at 1 am.

Wed 3 Oct **Meeting 9** at 11 am: For loops. *Read p. 67–73.*

Mon 8 Oct **Meeting 10** at 11 am: Loop exercises. **Assignment 5** due at 1 am.

Wed 10 Oct **Meeting 11** at 11 am: Compound statements, loop variable declarations. *Read p. 74–81.* **Quiz 3.**

Mon 15 Oct **Meeting 12** at 11 am: More loop exercises. **Assignment 6** due at 1 am.

Wed 17 Oct **Meeting 13** at 11 am: Break and continue. **Quiz 4.**

Mon 22 Oct **Meeting 14** at 11 am: Review session **Assignment 7** due at 1 am.

Wed 24 Oct **Midterm exam** at 11 am:

Mon 29 Oct **Meeting 16** at 11 am: Introduction to functions. *Read p. 83–93.*

Wed 31 Oct **Meeting 17** at 11 am: Local and global variables. *Read p. 93–106.*

Mon 5 Nov Meeting 18 at 11 am: Exercises with functions.

Wed 7 Nov Meeting 19 at 11 am: Recursive functions. *Read p. 106–116.*

Mon 12 Nov Meeting 20 at 11 am: Arrays. *Read p. 117–122.* **Quiz 6.** **Assignment 8** due at 1 am.

Wed 14 Nov Meeting 21 at 11 am: Randomness. *Read p. 123–128.*

Mon 19 Nov Meeting 22 at 11 am: Exercises with arrays. **Assignment 10** due at 1 am.

Mon 26 Nov Meeting 23 at 11 am: Arrays of arrays and two-dimensional arrays. *Read p. 128–144.*

Wed 28 Nov Meeting 24 at 11 am: Introduction to pointers. *Read p. 145–161.* **Quiz 7.**

Mon 3 Dec Meeting 25 at 11 am: Array processing using pointers. *Read p. 161–168.* **Assignment 11** due at 1 am.

Wed 5 Dec Meeting 26 at 11 am: String manipulation functions in C and C++. *Read p. 169–185.* **Quiz 8.**

Mon 10 Dec Meeting 27 at 11 am: Exercises with strings. *Read p. 185–195.*

Wed 12 Dec Meeting 28 at 11 am: Loose ends — overloading, do-while, switch statements, multiple modules. *Read p. 227–236.* **Assignment 12** due at 1 am.

Fri 14 Dec Final Exam at 11 am: Option 1

Wed 19 Dec Final Exam at 9 am: Option 2

Thu 20 Dec Final Exam at 11 am: Option 3