CS 164 Syllabus

18 January 2017

> A study of software project management concepts, software cost estimation, quality management, process involvement, overview of analysis and design methods, user interface evaluation, and design. Also considered are dependable systems – software reliability, programming for reliability, reuse, safety-critical systems, verification and validation techniques; object-oriented development; using UML; and software maintenance.

Welcome to CS 164. In this course, we will learn the process behind software development from front to back, by building a real project through the whole semester. You will be responsible for many parts of the system yourself, but we will discuss the overall design and direction as a class so that we can stay on track and learn from each other.

When: Monday, Wednesday 11am–12:50pm
Where: LLC 207
Credits: 3
Prerequisites: CS130

Contact Info

Instructor: Prof. Christopher League, Ph.D.
Email: `christopher.league@liu.edu` — please include the course number (CS164) in the subject. I have several email addresses, but all messages end up in the same place, so **please use only one.**
Google Hangout: `cleague@gmail.com`
Office hours: Monday, Wednesday 2:00–2:50, or make an appointment at `https://liucs.net/bookme`
Office phone: +1 718 488 1274
Office location: LLC 206, LIU Brooklyn

Resources

We will use several web resources:

- `https://liucs.net/cs164s17/` — notes, schedule, assignment handouts
- `https://gitlab.liu.edu/` — assignment submission, sample code
- `http://www.gradechamp.com/` — grade reports

There is no required textbook, but if you'd like a book to supplement or for reference, here are some suggestions:

- *The Pragmatic Programmer* by Andrew Hunt and David Thomas http://amzn.to/1AkWwHo
- *Team Geek* by Brian W. Fitzpatrick and Ben Collins-Sussman http://amzn.to/1OjkwnW or the newer version:
- *Debugging Teams: Better Productivity through Collaboration* by Brian W. Fitzpatrick and Ben Collins-Sussman http://amzn.to/1ZlFh5u
- *The Effective Engineer* by Edmond Lau https://www.theeffectiveengineer.com/book

Requirements

There are a total of 1,000 points available, broken down as follows:

- There will be 7 **project milestones** scheduled throughout the semester. The exact requirements and expectations for each will be posted to the course web site. Your contribution will be worth **125 points each,** but I will drop the lowest, so that only 6 milestones count, for a total of **750 points. Warning:** the *last* milestone cannot be dropped.
- There will be 7 **'check-in' opportunities** scheduled. These vary from week to week, but may involve responding to a survey, taking a brief online quiz, or participating in a discussion forum. Check-ins are worth **25 points each,** but I will drop the lowest two scores so only 5 will count, for a total of **125 points.**
- There is no midterm exam, but there will be a final exam, worth **125 points.**

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 achieve 930 points, you are guaranteed an **A**.

Policies

It is important to **complete tasks on time,** so you don't fall behind. Late work will be graded as follows.

This formula specifies a *lateness factor* f that is multiplied by your earned score to

determine a late score. The variable h represents the number of hours the submission is late.

$$f = \frac{8.5 - \log_2\left(\frac{h}{24}\right)}{10}$$

There will be no extra credit. Students usually ask for extra credit late in the semester after they have already squandered 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 https://liucs.net/u2

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.

In a course with programming assignments, it is usually okay to work with and learn from other students to **some** extent, but what you submit in the end needs to be your own. The most reliable way to do that would be to set aside whatever code you created together, and then recreate it from scratch on your own.

Showing up on time to class 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,** whether psychological, neurological, chronic medical, learning, sensory, or physical. The 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 Student Support Services as early as possible and

to provide faculty members with the formal communication for suitable accommodations. Visit Pratt 410, call 718 488 1044, or visit http://www.liu.edu/Brooklyn/SSS

I participate in the **LIU Safe Zone** program. Representatives of the program serve as contacts for individuals on campus with questions or concerns related to sexual orientation and gender identity, whether of self or of a friend or family member. The goal of the program is to promote a safe and free campus for all students. Safe Zone areas can be identified by a sticker with the LIU Safe Zone logo.

The **Family Educational Rights and Privacy Act** (FERPA) gives students control over the disclosure of their educational records. During this course you may have the opportunity to create accounts or register with certain public online services. In these cases, you need not make any personally identifying information public. You may use a pseudonym or online handle, as long as you identify yourself to the instructor.

Time commitment

New York State defines one credit as a total of 15 hours instructional time, plus 30 hours of student preparation. Thus, a typical three-credit course will amount to 45 hours instruction plus 90 hours preparation. (For these computations, an 'hour' actually consists of 50 minutes.)

To perform well, you will have to spend some time preparing and reviewing outside of class, and a **significant** amount of time completing programming assignments (keeping in mind that earlier assignments will require less time than later ones).

- Lecture time: 4 hours per week × 15 weeks = 60 hours
- Preparation time (reading, reviewing): 2 hours per week × 15 weeks = 30 hours
- Assignment completion (problem-solving, programming): approximately 10–14 hours per assignment × 7 assignments = 90 hours.
- **Total: 180 hours**

Goals and objectives

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

- demonstrate proficiency in basic algorithms and data structures (1.1, mastery level).
- understand the mathematical and logical foundations of computing (1.2, mastery level).
- master the fundamental facilities of various programming languages and software architectures (2.1, mastery level).
- effectively use tools for software development (2.2, mastery level).
- develop a data modeling design for a proposed database application (3.2, mastery level).

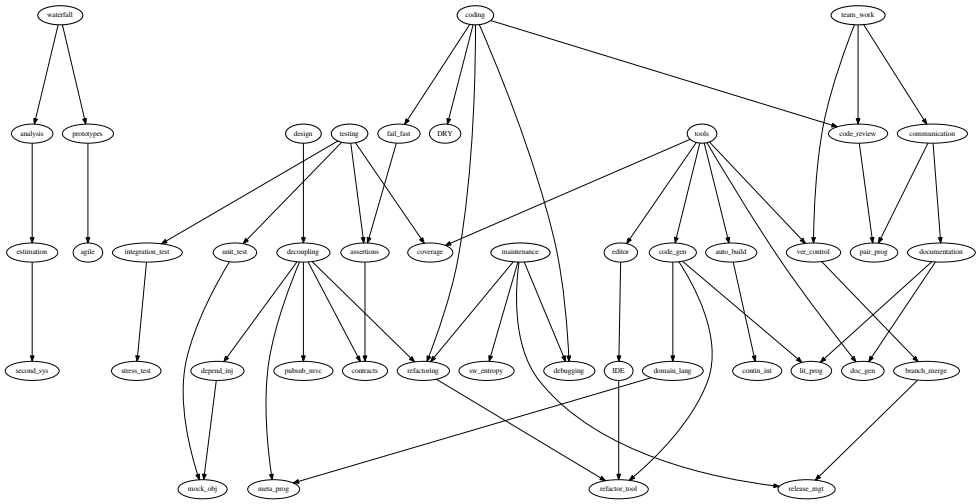

Figure 1: Topics and their dependencies – see full size image on web site

- communicate technical ideas and specifications in writing (4.1, introductory level).
- give an effective oral presentation on some technical subject area (4.2, introductory level).
- exhibit awareness of professional organizations and technical opportunities (5.1, mastery level).
- productively attend seminars and workshops outside of class work (5.2, mastery level).

Schedule

We will cover these areas of the software development life-cycle:

1. Requirements analysis and project planning
2. Software architecture and system design
3. Implementation tools and techniques
4. Verification and validation
5. Deployment and maintenance

The day-by-day schedule is shown below, including all deadlines.

Wed Jan 18 Meeting 1 at 11 am. Software Development Lifecycle (Waterfall)
Mon Jan 23 Meeting 2 at 11 am. Mythical man-month, second-system syndrome, prototyping. Perceptrons.
Tue Jan 24 Check-in 1 due at 23:59.
Wed Jan 25 Meeting 3 at 11 am. Perceptron learning rule, Python doctest module.

Mon Jan 30 Meeting 4 at 11 am. Demonstrate OpenAI universe, implement perceptron learning.
Wed Feb 1 Meeting 5 at 11 am. Version control.
Sun Feb 5 Milestone 1 due at 23:59. **Check-in 2** due at 23:59.
Mon Feb 6 Meeting 6 at 11 am. Git hashes and internal representation.
Wed Feb 8 Meeting 7 at 11 am. Adapt perceptron to hill-climbing, apply to cart-pole.
Mon Feb 13 Meeting 8 at 11 am. Deeper recap on perceptron model, what is gym doing?
Wed Feb 15 Meeting 9 at 11 am.
Sun Feb 19 Milestone 2 due at 23:59.
Tue Feb 21 Meeting 10 at 11 am. Introducing lunar lander problem, matrix operations, and numpy.
Wed Feb 22 Meeting 11 at 11 am. More tool and techniques for unit-testing in Python, including unittest and coverage modules.
Mon Feb 27 Meeting 12 at 11 am. Property-based testing and fault injection.
Tue Feb 28 Check-in 3 due at 23:59.
Wed Mar 1 Meeting 13 at 11 am.
Mon Mar 6 Meeting 14 at 11 am. Reinforcement learning overview.
Tue Mar 7 Milestone 3 due at 23:59.
Wed Mar 8 Meeting 15 at 11 am. Implementing and optimizing k-bandit agents.
Sun Mar 19 Check-in 4 due at 23:59.
Mon Mar 20 Meeting 16 at 11 am. Delta-rule approximation vs exact averages in k-bandit agents, and the recency effect.
Wed Mar 22 Meeting 17 at 11 am. Value function calculation for grid world problems.
Mon Mar 27 Meeting 18 at 11 am. Specified a new, finite grid world problem for M4.
Wed Mar 29 Meeting 19 at 11 am. Assertive programming, design by contract, and Hoare logic.
Sun Apr 2 Milestone 4 due at 23:59. **Check-in 5** due at 23:59.
Mon Apr 3 Meeting 20 at 11 am. Policy improvement for solving grid world of M4.
Wed Apr 5 Meeting 21 at 11 am. Tiling a continuous space
Sun Apr 9 Milestone 5 due at 23:59.
Mon Apr 10 Meeting 22 at 11 am. Apply tiling to cartpole
Wed Apr 12 Meeting 23 at 11 am. Training multi-layer perceptrons with scikit-neuralnetwork
Sun Apr 16 Check-in 6 due at 23:59.
Mon Apr 17 Meeting 24 at 11 am. Minesweeper
Wed Apr 19 Meeting 25 at 11 am. More minesweeper coding, using constraints
Sun Apr 23 Milestone 6 due at 23:59.
Mon Apr 24 Meeting 26 at 11 am. Start on a constraint solver
Wed Apr 26 Meeting 27 at 11 am. Solution to M6, plugging constraints into Minesweeper

Sun Apr 30 **Check-in 7** due at 23:59.
Mon May 1 **Meeting 28** at 11 am.
Sun May 7 **Milestone 7** due at 23:59. Final submission of all project materials.
Mon May 8 **Final exam** due at 23:59.