EE355x – Software Design for Electrical Engineers
Nazarian Spring 19
University of Southern California
Department of Electrical Engineering

This is the first Draft. May have a major update soon.

Course Description

This 4-unit course offers the subjects of computer science and software development relevant to electrical engineering applications. The ability to model, simulate, analyze, characterize problems and their solution space, and efficiently verify the design, is critical in the modern practice of engineering. To prepare students to carry out these tasks, this course will include object-oriented programming using C++, as well as scripting using Python. Focus will also be placed on relevant aspects of algorithms, discrete mathematics, data structures, computational complexity and debugging skills. An interactive lecture/lab/discussion course format will be utilized to provide hands-on experience and active learning techniques. Students who complete this course will be prepared for entrance level industrial jobs and more advanced level courses in the areas of computer science and engineering (such as the Spring 18 EE599 – Special Topics: Software Design and Optimization for Electrical Engineers).

Website

https://blackboard.usc.edu/webapps/login/

Instructor

Shahin Nazarian
Contact: http://atrak.usc.edu/~shahin/, shahin.nazarian@usc.edu
Office Hours: Check BB.

Teaching Assistants and Graders

Class Schedule Information

https://classes.usc.edu/term-20181/classes/ee/

Grading Policy

HW Assignments	5%
Lab Assignments	15%
Programming Assignments (PAs)	20%
Programming Exam	5%
Written Midterm I, Midterm II	Highest of (35%, 20%) and (20%, 35%)

Academic Integrity

Plagiarism, cheating and unauthorized collaborations will not be tolerated. You should either attend the first lecture and/or watch the recorded lecture for more details regarding the assignments and our course policies. Please also ask the instructor if you have questions about proper behavior and/or review the information (including a 15-minute tutorial) posted in the following website:
http://www.usc.edu/libraries/about/reference/tutorials/academic_integrity/index.php
More information about academic integrity including USC policies can be found at the following:
http://www.usc.edu/student-affairs/SJACS/pages/students/publications.html

Learning Objectives

Upon completion of this course students will be able to:
1. Understand the principles of program execution on modern computing hardware including memory management, subroutine calls, and efficiency issues.
2. Model a given problem as a set of objects and define their interaction.
3. Understand and implement basics of verifiability.
4. Use object-oriented features of C++ to solve problems.
5. Utilize appropriate data structures from the C++ Standard Template Library (STL)
6. Understand principles of discrete mathematics related to sets, graph theory, and certain numerical methods.
7. Use software development tools and packages including debuggers, makefiles, and source code management (version control) tools.
8. Use Python to perform basic scripting tasks.
9. Understand threading models and basic synchronization primitives.
10. Perform a non-trivial software design project from conception to implementation

Prerequisite(s)	EE155 or ITP165
Co-Requisite(s)	N.A.
Concurrent Enrollment	N.A.
Recommended Preparation	N.A.

Technological Proficiency and Hardware/Software

Any coding experience may directly or indirectly help.

Required Readings and Supplementary Materials

- Lecture slides: will be posted on the course webpage
- Lecture notes: your responsibility to take them

- Supplementary material including research papers, and technical articles (e.g., on machine learning) will be provided.

Additional (Recommended) Readings

- Main Textbook: Cay Horstmann, C++ for Everyone, 2nd Ed. Wiley Publishing, 2010. ISBN: 978-047092713-7.
- http://cplusplus.com/doc/tutorial/
- http://www.doc.ic.ac.uk/~wjk/C++Intro/
- http://c.learncodethehardway.org/book/
- Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2nd Edition, 2016, ISBN: 978-1491939369.

Exams

There are two written exams, both of which are closed book; with no electronics allowed. Part of each exam is dedicated to directly testing students on the concepts of lectures/hw/lab/prog. assignments. There is also one programming test. You will see the guidelines to each exam about 5 to 7 days in advance to the exam date. Any cheating may result in an "F" in the course and will be referred to Student Affairs for other penalties.

HW, Lab, and Project Assignments

- Assignments are your key to learning. They are designed to familiarize you with the problems and skills you will need during interviews and also an entry level job in computer science or engineering fields.
- Homework assignments support the theory coverage of our course.
- Practicing the actual coding in lab/programming tasks on your own will you help you develop the skills and understanding to succeed.
- Each assignment may be assigned in parts to help the students with time management. The final programming assignment will act as the final project.
- You will find the submission information in the document header.
- You should write your first name, followed by your last name, and your student ID on the first page of your submission (first page of HW, and readme.pdf file of your lab/project assignments). You will lose at least 5% of the total points if any of the afore-mentioned items is missing, or the header info is not followed.
- It is expected that you will present your own work in your own creative way.
- No collaboration (no discussions are allowed among students) is allowed on the individual assignments.
- If you do not know the answer to a problem in any assignment you are encouraged to use our support (office hours, blackboard discussion board, etc.) It is wiser to not submit and lose some points than facing the academic dishonesty consequences.
- **We take academic honesty very seriously. Powerful techniques are used to detect whether a student copied from others in class, previous semesters or online resources. However, despite all the warnings, some students are caught violating the academic integrity rules, every semester. Those caught discussing, copying, allowing others to copy, or any other type of unauthorized collaboration or plagiarism on any lab or project assignment will be reported to USC Student Judicial Affairs and Community**

Standards (SJACS). The minimum penalty will be a 0 on the assignment and one letter grade reduction for the entire course. However, depending on the case more severe penalties such as failing the course and expulsion from USC may be applied by the SJACS. In addition to possible financial consequences, a record of academic dishonesty can seriously damage one's USC standing or future career.

- We highly recommend posting your questions on the blackboard discussion board for each assignment, so others would be able to review your questions and our answers.
- All emails to the TAs and graders, regarding the assignments, should be cc'ed to shahin.nazarian@usc.edu.

Course Attendance

The lecture and discussion sessions are interactive and demand in-class programming, discussion and research activities. Attendance is therefore mandatory, i.e., the students are not allowed to miss any of the lectures or discussions, unless there is a family or medical emergency.

Expectations

Students are expected to follow the academic honesty policies of USC, attend classes, take notes, and participate in discussions by asking questions and providing answers/arguments. You will be expected to have reviewed the posted slides before each lecture. This will greatly aid your comprehension of the lecture material.

Statement on Academic Conduct and Support Systems

Academic Conduct

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Section 11, Behavior Violating University Standards (https://scampus.usc.edu/1100- behavior-violating-university-standards-and-appropriate-sanctions/). Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct (http://policy.usc.edu/scientific-misconduct/). Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the Office of Equity and Diversity (http://equity.usc.edu/) or to the Department of Public Safety (http://capsnet.usc.edu/department/department-public-safety/online-forms/contactus). This is important for the safety whole USC community. Another member of the university community - such as a friend, classmate, advisor, or faculty member - can help initiate the report, or can initiate the report on behalf of another person. The Center for Women and Men (http://www.usc.edu/student-affairs/cwm/) provides 24/7 confidential support, and the sexual assault resource center webpage sarc@usc.edu describes reporting options and other resources.

Support Systems

A number of USC's schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the American Language Institute http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. The Office of Disability Services and Programs http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, USC Emergency Information http://emergency.usc.edu/ will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.

TENTATIVE COURSE SCHEDULE and TOPICS
[Tentative. Dates, etc. Will Be Updated]

Wk	Monday Lecture	Monday Lab	Wednesday Lecture	Wednesday Quiz	Recommended Textbook Sections
1	1/7: Syllabus, and Introduction, Python	1/7: Setup, Prelab, Lab1 assigned	1/9: Python (cont.)	1/9: Lab1 HW1 assigned	Ch.1 & 2
2	1/14: Algorithms and Complexities (cont.)	1/14: **Prelab/HW1 due** HW2/Lab2 assigned	1/16:	1/16: **Lab1 due**	Ch. 5 & 12
3	1/21: University Holiday (MLK Day)	1/16:	1/23: OOP Fundamentals Functions, Arrays	HW3/Lab3/ PA1 assigned **HW2 due**	Ch. 9 & 10
4	1/28: OOP Fundamentals (cont.)	1/29: HW4/Lab4 assigned **HW3 due**	1/30: C Review (Data Types and Operators	1/30: Lab2 assigned **Lab3 due**	Ch. 3 & 4
5	2/5: C Review (Control, Loops, Function)	2/5: HW5/PA2 assigned **HW4 due**	2/7: I/O **PA1 due**	2/7: **Lab4 due**	6.1,6.2.1, 6.2.2
6	2/12: Pointers	2/12: HW6/Lab5 assigned **HW5 due**	2/14: Pointers (cont.), References	**PA2 due (on 16th)**	4.6.1,4.6.2, 6.3
7		**HW6 due (on 20th)**		**2/21: Prog. Test I**	12.1,12.2
8	2/26: References (cont.), Arrays	HW7/PA3 assigned	2/28: Arrays (cont.)	2/28:	4.4,4.5

		Lab5 due			
9	3/5: Strings	HW8/Lab6 assigned **HW7 due**	3/7: Strings (cont.)	**3/7: Written MI**	9.2.4 **PA3 due (on 9th)**
10	3/19: Dynamic Memory Allocation	3/19: Lab7/PA4/HW9 assigned **HW8 due**	3/21: Dynamic Mem. (cont.)	3/21: **Lab6 due**	5.1, 5.2
11	3/26: STL Lists (Vector, Deques)	3/26: HW10/Lab8 assigned **HW9 due**	3/28: List Lists	3/28: **Lab7 due**	5.3
12	4/2: Doubly Link Lists (cont.)	4/2: HW11/PA5/Lab9 assigned **HW10/Lab8 due**	4/4: Recursion	4/4: **PA4 due**	10.2.2-10.2.5
13	4/9: Maps, Sets	4/9: Lab10 assigned **HW11 due**	4/11: Parallel Programming	**4/11: Prog. Test II**	11.1-11.4
14	4/16: Machine Learning	4/16: PA6/HW12 assigned **Lab10 due**	4/18: Alg. (cont.)	4/18: MII Practice **PA5 due**	11.5-11.11
15	4/23: Alg. (cont.)	**HW12 due**	4/25: Alg. (cont.)	**4/25: Written MII**	13.3,13.4
				5/2: **PA3 due**	