CSCI 201L Syllabus
Principles of Software Development
Fall 2018

Instructor: Jeffrey Miller, Ph.D.
Email: jeffrey.miller@usc.edu
Web Page: http://www-scf.usc.edu/~csci201
Office: SAL 342
Phone: 213-740-7129

Lectures:
30303R, Tuesday/Thursday 8:00a.m.-9:20a.m., SLH 102
29909R, Tuesday/Thursday 9:30a.m.-10:50a.m., GFS 116
30245R, Tuesday/Thursday 11:00a.m.-12:20p.m., GFS 116

Quiz:
30028R, Thursday, 7:00p.m.-8:50p.m.

Labs:
30237R, Tuesday, 2:00p.m.-3:50p.m., SAL 126
30134R, Tuesday, 4:00p.m.-5:50p.m., SAL 126
30241R, Tuesday, 6:00p.m.-7:50p.m., SAL 109
30396R, Tuesday, 6:00p.m.-7:50p.m., SAL 126
30239R, Wednesday, 10:00a.m.-11:50a.m., SAL 126
30238R, Wednesday, 12:00p.m.-1:50p.m., SAL 127
30385R, Wednesday, 4:00p.m.-5:50p.m., SAL 109
29904R, Wednesday, 6:00p.m.-7:50p.m., SAL 109

Office Hours:
Tuesday 6:00a.m.-7:00a.m. and 12:30p.m.-1:30p.m.
Thursday 6:00a.m.-7:00a.m. and 1:30p.m.-3:30p.m.
Any day by appointment

Textbooks:
Liang, Y. Daniel. Introduction to Java Programming, Comprehensive Version, 11th Edition, Prentice Hall, Inc., 2017. ISBN 978-0134670942

Description:
Object-oriented paradigm for programming-in-the-large in Java; writing sophisticated concurrent applications with animation and graphical user interfaces; using professional tools on team project.

Grades:

Labs	10%	Assignments	20%
Written Exam #1	15%	Group Project	30%
Written Exam #2	15%	Lecture Attendance	10%

Grades will be based on a curve that operates in favor of the students, with at least the following grades for a given percentage x. If the average in the class is lower than 80%, the average will become the cut-off between a B- and a C+.

x >= 93	A	73 <= x < 77	C
90 <= x < 93	A-	70 <= x < 73	C-
87 <= x < 90	B+	67 <= x < 70	D+
83 <= x < 87	B	63 <= x < 67	D
80 <= x < 83	B-	60 <= x < 63	D-
77 <= x < 80	C+	x < 60	F

Lecture Schedule: Chapter references are from Y. Daniel Liang, Introduction to Java Programming and Data Structures, Comprehensive Version, 11th Edition, Prentice Hall, 2017. ISBN 978-0134670942

Week	Lecture	Date	Lecture Topic	Lab Topic	Chapter
1	1	August 21, 2018	Introduction, Environment, Methods	Github Tutorial, Environment Setup	1-8
	2	August 23, 2018	Classes, Packages, File I/O		9-10
2	3	August 28, 2018	Abstract Classes and Interfaces, Inheritance, Polymorphism	Inheritance	11, 13
	4	August 30, 2018	Garbage Collection, Exception Handling, Serialization, Generics, Inner Classes		12, 19
3	5	September 4, 2018	HTML, CSS	HTML, CSS	
	6	September 6, 2018	Java Servlets, JSP		37-38
4	7	September 11, 2018	JavaScript	JSP and Servlets	
	8	September 13, 2018	AJAX		
5	9	September 18, 2018	Concurrent Computing	JavaScript and AJAX	32
	10	September 20, 2018	Thread Methods, Thread Pools, Thread Priorities		32
6	11	September 25, 2018	Software Engineering, Methodologies, Testing	Threads	
	12	September 27, 2018	Project Description		
7	13	October 2, 2018	Networking Theory	Software Engineering	33
	14	October 4, 2018	Networking Theory (cont.)		33
8	15	October 9, 2018	Network Programming	Networking Worksheet	33
	16	October 11, 2018	Multi-Threaded Network Programming		33
		October 11, 2018 7:00p.m.-8:50p.m.	**Written Exam #1**		
9	17	October 16, 2018	Network Serialization	Web Server	
	18	October 18, 2018	Web Sockets		34-35
10	19	October 23, 2018	Databases	MySQL Installation	34-35
	20	October 25, 2018	SQL		34-35
11	21	October 30, 2018	JDBC	JDBC	32
	22	November 1, 2018	Concurrency, Monitors		32
12	23	November 6, 2018	Locks, Conditions, Producer/Consumer	Locks and Monitors	32
	24	November 8, 2018	Sleeping Barber, Locks Programming		32
13	25	November 13, 2018	Multi-Threaded Programming Design, Semaphores	Sleeping Barber	32
	26	November 15, 2018	Parallel Computing		40
14	27	November 20, 2018	Project Demonstrations	*No Lab*	
		November 22, 2018	**No Class - Thanksgiving**		
15	28	November 27, 2018	Project Demonstrations	Parallel Computing	
	29	November 29, 2018	Project Demonstrations		
16		**December 8, 2018 11:00a.m.-1:00p.m.**	**Written Exam #2**		

Exams:

Written Exam #1	Thursday	October 11, 2018	7:00p.m.-8:50p.m.
Written Exam #2	Saturday	December 8, 2018	11:00a.m.-1:00p.m.

The written exams are closed book and will consist of theoretical questions and may have code to be analyzed, though very little code will be required to be written. You are allowed one 8.5"x11" of double-sided **hand-written** notes for the exam, but no other resources are permitted.

An exam can only be taken on the scheduled date and at the scheduled starting time. Accommodations for students with letters from DSP will be provided, though the exam will still need to be taken on the scheduled date. There are no makeup exams. If you miss an exam due to an emergency, official written documentation, whatever that may be based on the situation, will need to be submitted to me as soon as you are physically able (before the exam if possible). Approval will be solely based on my discretion though it should be based on a documented illness or emergency. Based on the exam, here are the rules that will be followed:

- If an excuse is not approved, you will be given a 0 on the exam.
- If there is an approved excuse for written exam #1, the percentage for that exam will be added to the percentage for written exam #2.
- If there is an approved excuse for written exam #2, you will receive an Incomplete grade in the course and have to make up the exam based on the conditions of an Incomplete.

Assignments: Assignments will be discussed in class and worked on individually. Discussion among students is fine, but no copying of other student's code is allowed. The program needs to compile, and grading will only occur if the program is able to be run. Assignments will be submitted via Github or Blackboard (instructions will be provided in class) and are due by 11:59p.m. on the due date (see Late Policy below). Grading criteria will be provided with the assignment description. Graders will grade the assignments. Once grades are entered into Blackboard, students will be able to request a regrade if they think a mistake has been made in the grading through the following process:

1. Within five days of receiving the grade on an assignment, submit a formal request with the online form. Note that this is the only request that can be made to regrade that specific assignment, so be sure to include all relevant information. If the request is submitted more than five days after the grades are posted, the request will be denied.
2. The TA will review the request and determine if a regrade will be granted.
   a. If the regrade request is denied, the original grade will stand.
   b. If the regrade request is granted, the TA will forward the request to a grader (possibly a different one than who originally graded it). The grader will conduct a regrade and send the updated grade to the TA, who will then enter it into Blackboard.
3. There will only be one regrade request, and the grade after the regrade is final. If any questions arise beyond that, the student will need to speak with the professor in person.

Project: The project in the class will be assigned approximately half-way through the semester. The project will consist of between 4-6 students. Formal documentation following the software engineering process will be required. The project will be discussed in class with the corresponding due dates. The project deliverables will be submitted via Blackboard and is due by 11:59p.m. on the due date (see Late Policy below).

Late Policy: There is no late policy. For any assignment or project that is submitted after 11:59p.m. on the due date, the student will receive a 0. Assignments should be submitted early in case some situation arises that prohibits a student from submitting an updated version before the deadline. I understand that things happen, students get sick, accidents occur, computers crash, and so on, so budget your time appropriately taking into account any risk factors.

Labs: The TA/CPs will lead the lab section each week. There will be an assigned lab program each week that reinforces the topics covered in the lectures. The lab assignments will be graded based on effort and attendance. You should NOT complete the lab before your assigned lab section. The labs are intended to be completed during the lab period, and you are expected to work on the lab during the section. The lab assistants have been instructed NOT to grade any labs until after half of the lab has elapsed. You also must show up to the lab within the first 10 minutes or you will not be allowed to join the lab that week. Solutions will be posted to the labs, but you should make sure you are using the lab periods to learn the material that has been covered. The lab assistants are there to answer any questions and help you, so use your time in lab wisely. Each lab is worth 0.8% of the final grade, and the total lab score is out of 10%.

Attendance: Attendance will be taken during lecture. To receive full credit for attendance, you must attend 26 lectures out of the 29 lectures this semester. Because there are three lectures that can be missed without penalty to your final grade, there are **no** excused absences. You can miss three lectures for whatever reason, but no student will be given attendance points without attending the lecture. If a lecture is cancelled for any reason, it does not change that you still must attend 26 lectures over the semester to receive full credit for attendance.

To earn full credit for the lecture, you must arrive within the first 10 minutes of class. Arriving after 10 minutes will not earn credit.

Prerequisites: CSCI 104L – Data Structures and Object Oriented Design

Students with Disabilities: Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to the professor as early in the semester as possible.

Academic Integrity:

The Viterbi School of Engineering's policy on Academic Integrity can be found at http://viterbi.usc.edu/academics/integrity/. All students are expected to understand and abide by these principles. SCampus (http://scampus.usc.edu), the Student Conduct Code, contains the information about violating University standards in Part B Section 11. Any potential violations will be taken seriously and the proper academic process will be followed, including reporting to the USC Student Judicial Affairs and Community Standards (SJACS).

Viterbi Honor Code:

Engineering enables and empowers our ambitions and is integral to our identities. In the Viterbi community, accountability is reflected in all our endeavors.

Engineering+ Integrity.
Engineering+ Responsibility.
Engineering+ Community.
Think good. Do better. Be great.

These are the pillars we stand upon as we address the challenges of society and enrich lives.

Course Outcomes (expected after you've finished the course)

i	The ability to understand the software engineering in terms of requirements, design, and implementation;
ii	An understanding of how to use interaction diagrams to help define requirements;
iii	The ability to produce a software design based on requirements;
iv	The ability to produce software, including graphical user interfaces, from a design;
v	The ability to unit test a module;
vi	An understanding of concurrency and how it works in computer operating systems;
vii	The ability to write multi-threaded programs and correctly solve a mutual exclusion problems using semaphores or monitors;
viii	The ability to use Java in writing programs;
ix	The ability to use HTML and CSS in designing graphical user interfaces;
x	The ability to use messaging as a communication method;
xii	The ability to apply a software engineering process to a large software project;
xiii	The ability to work effectively on a team;
xiv	An understanding of the ethical issues in working within a group;

ABET Program Outcomes (after you have finished your degree)

a	An ability to apply knowledge of computing and mathematics appropriate to the discipline;
b	An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
c	An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
d	An ability to function effectively on teams to accomplish a common goal;
e	An understanding of professional, ethical, legal, security, and social issues and responsibilities;
f	An ability to communicate effectively with a range of audiences;
g	An ability to analyze the local and global impact of computing on individuals, organizations and society;
h	Recognition of the need for, and an ability to engage in, continuing professional development;
i	An ability to use current techniques, skills, and tools necessary for computing practices.
j	An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
k	An ability to apply design and development principles in the construction of software systems of varying complexity.