SYLLABUS

COMP 110/110L Intro to Algorithms and Programming Fall 2017

Instructor: Dr. Jeff Drobman
Dr Jeff's Web Page: http://www.drjeffsoftware.com
Course Web Page: http://www.drjeffsoftware.com/classroom.html
Email: jeffrey.drobman@csun.edu
Office Hours: In Lab, immediately after class

Course Description

Introduction to algorithms – their representation, design, structuring, analysis and optimization – as a means to solving problems and expressing computer applications. Implementation of algorithms as structured, object-oriented programs in the high-level application programming language *Java*. Some comparisons to other languages such as C and C++. We will be applying to a series of lab programs and projects the paradigms of *Software Development Life Cycle* (SDLC) and *Input-Process-Output* (IPO).

Goals

This course teaches basic skills in analyzing problems, and solving them by finding or creating an appropriate *algorithm* (a recipe for solving a problem). Once an algorithm has been identified, it is translated into an executable application using the Java programming language. This last task is often called "programming," and it should follow best practices for "software engineering".

This encompasses the steps called "Design" and "Implementaton" in the SDLC. Additionally, all computer application programs must satisfy the set of prescribed requirements (given by the instructor in this case) – the first step or "stage" in the SDLC. Lastly, they must be fully tested and debugged (4th stage of SDLC).

This course does not assume any previous experience in computer programming, and material begins at an introductory level. However, coverage is fast-paced and moves on to more advanced topics quickly. This is not a survey course for non-majors; it is a real programming course designed for students concentrating in Computer Science and related majors who need to quickly develop real programming skills.

Java

The principles and skills related to problem solving and algorithms are general and are not specific to any programming language. However, it is imperative to also learn the modern programming paradigm (model) of object-oriented design (OOD) – which is at the core of the language Java. Most client-side computer applications today are written in either Java or C++ (or its derivatives).

The CSUN CS Department has made a policy decision to use the Java language for lower division courses such as COMP 110, so Java language details will be presented as an important part of both lecture and lab, although the policy may change in the future. For now, you must demonstrate an appropriate level of both problem solving and Java programming skill to successfully complete the course. The lecture component (COMP 110) focuses on concepts and practical examples. The lab (COMP 110L) focuses on developing working and correct applications using Java, with attendant problem solving, program debugging and error handling techniques.

Textbook

There is one required textbook: Y Daniel Liang, Introduction to Java Programming, Brief Version, 10th or later edition, Prentice Hall. It is good practice for students to bring their textbook to class. The instructor will make references to various pages and sections in this book from time to time.

Exams

Exams, quizzes, programming assignments ("labs") and term projects are based on what is presented in the lecture. There will be one Quiz (at first quarter), a Midterm and a Final Exam. All these exams will have one part that is multiple choice entered on Scantron forms. The Midterm and Final Exams will have an additional section for writing code segments to demonstrate capability.

Assignments

For programming assignments, here will be 8 continually scheduled "Labs" plus two term projects – one for each half term. The labs are carefully selected to integrate the current lecture material, and lead the student to a mastery of traditional and fundamental programming techniques, areas and algorithms.

The two "projects" are each designed to fully incorporate all the material learned up to the mid-point (#1) and over the entire course (#2). They are opportunities for the student to demonstrate their mastery of this course.

All programming assignments will be accompanied by "starter code" provided by the instructor – essentially "skeleton" code. Students will need to fully understand the provided code, and then add to it the student's own code for completion. Students will be graded on the part of the code that they produce.

Submissions of labs and projects will use a provided form for documentation to include each student's input, output and source code listing. All assignments are to be submitted as files attached to emails sent to the instructor directly (using my csun.edu address above).

Collaboration

Collaboration among students is encouraged. However, this must fall short of mere copying code, which is plagiarism. (That said, it is often the case that there is only one good solution to each part of an assignment, and this instructor will take that into account.)

Late Submission Policy

All assignments will have a due date. You will have a one week grace period. After that, late assignments become "past due" and are assessed a "late penalty" of 10% per week – for 2 more weeks. No assignments will be accepted after that final past due period. Finally, all remaining assignments (if any) must be submitted by the last day of class (regular classes, not the final exam week). No assignments will be accepted after the last day of class.

Lecture Material

All lectures will be delivered via PowerPoint slides (as PDF files). All slide sets will be updated and posted in PDF format to the class web page. However, it is recommended that for the best level of preparation, you should attend lecture and participate in discussions, rather than simply reading the lecture notes on your own. There will also be a limited amount of hand-drawn notes on the board, for which each student is responsible for taking notes.

Lab

Programming in Java is practiced in each lab session, putting to use concepts and constructs learned in the preceding lectures. The Java "SDK" called ***JDK***, from Oracle, is used in concert with the "IDE" called ***jGRASP*** (freeware from Auburn University) – on student workstations running Linux (Red Hat), and optionally on student owned PCs or Macs.

Learning to use the jGRASP IDE is an important part of the lab and the course. All source code editing, compiling, running and debugging will be done with this IDE. You will be shown how to utilize some of the extra features of it, such as generating *CSD* (Code Structure Diagram) and *UML* class diagrams.

The lab sessions will mostly involve the instructor going over the assignments, and then providing and running the "starter code" for each. Some related general code segments, and some other example code, may also be presented during the labs. During the lab sessions, students are encouraged to work on their assignments, and may ask for help from the instructor or the in-class tutor, or any other students.

It will usually be necessary to continue to work on your programming assignments on your own time outside of your official lab time. You will also need to work on your Term Project throughout the entire term, as part of your lab work.

Tutors

The School (CECS) provides a group of tutors available M-F in the designated tutoring room in Jacaranda Hall. Students will be expected to make use of this resource. There may also be a designated in-class tutor available during the lab sessions.

Grading

You will receive a **single combined grade** for both 110 and 110L. **Plus/Minus** grading will be used, as shown.

Grade	Pct	Interpret
A+	98	VERY good
A	**92**	
A-	90	
B+	88	PRETTY good
B	**82**	
B-	80	
C+	78	BARELY good
C	**72**	
C-	70	
D+	68	substandard
D	**62**	
D-	60	
F	<60	failed

	Category	Weight	
8 @5 ea	Programs	40	Programming 65
2 @ 10, 15	Project	25	
	Quiz	5	Testing 35
	Midterm	10	
	Final	20	

Single COMBINED GRADE

Computer Accounts

Every student registered for the course has a networked account that can be used on all CS Department computers. Your account (user id and password) is the same as the one issued by the University. The instructor does not have sysadmin authority over student accounts. For problems logging in, see the CECS Information Services office in Room JD 1112, extension 3919.

Software Tools

You may do your work on any Lab machine or on your personal computer (PC or Mac). Lab machines use the Linux OS, but the software applications required for class work are available for all common OS'es. The Java JDK and jGRASP come pre-installed on the lab machines. For your own computers, you will have to install these 2 tools. The tutors can help you with this.

When installing, it is important that you install the JDK first, then the jGRASP IDE. Make sure you install the Java "JDK", not the "JRE" (Java Runtime Environment), and the "SE" version (not the "EE" versions).

For the JDK, go to:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Download the Java SE (Standard Edition) JDK – latest version is "Java SE 8u<nnn>", where <nnn> is at least 144 now.
Install JDK first, before proceeding to download and install JGrasp.

For jGRASP, go to:

http://www.jgrasp.org

Click on the Downloads link and download the installation for your platform (Windows, MacOS, or Linux/Unix).

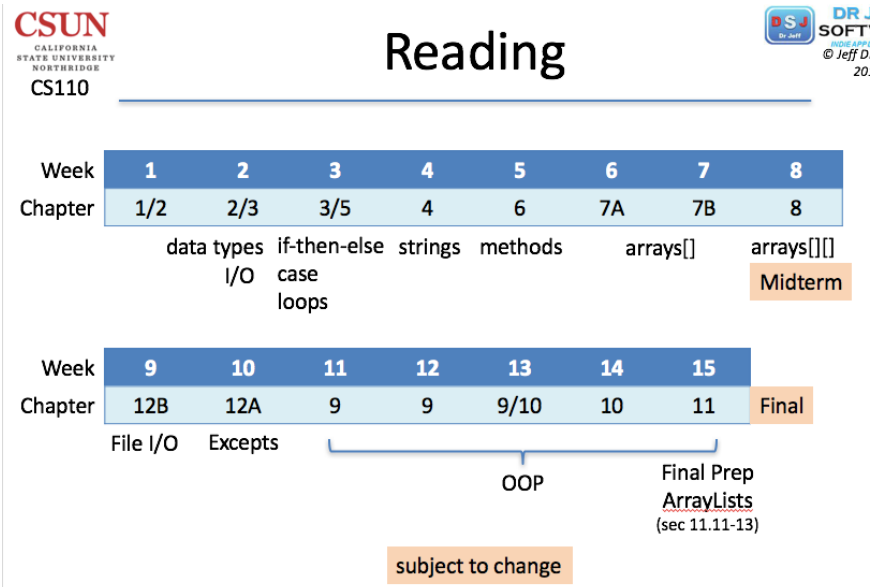
Exam Materials

We will use Scantron forms (long forms 882-E, not the short ones) for all exams, so bring one plus a No. 2 pencil to the exams.

Save Your Work

Work done in the lab can be saved to your personal directory on the CECS file server (the "Z drive"). Do not store anything of importance on the hard drive of the local machine, since the machines can be reformatted at any time without prior notification. In addition to Z drive storage, you can use a USB flash drive to store backup copies of work performed in the lab, or you can send an email to yourself with programming source code as an attachment.

Schedule of Topics (subject to change)



Attendance

For at least the first several weeks, attendance will be closely monitored in both lecture and lab. Frequent unexcused absences will result in a grade penalty. After the first few weeks of the semester, attendance will not be monitored as closely, but note that the scheduled lecture and lab time is the most appropriate time to ask questions and get help on your projects. I am not willing to spend an unreasonable amount of time outside of class (including excessive use of email) to explain class material to anyone who is not attending lecture and lab. In other words, you are expected to participate in lecture discussions by attending lecture, and get programming help on your projects in person by attending lab. Students are welcome to bring personal laptops with a wireless connection to both lecture and lab.

Personal Behavior in Lecture and Lab

You are expected to arrive to both lecture and lab on time, and to wait until dismissed before leaving. You are expected to be polite, pay attention, and participate in discussions: no web surfing, no private conversations, no cell phone interruptions, no sleeping/snoring, etc. You may quietly excuse yourself for bathroom breaks, to take important phone calls, and to address other urgent personal business, without asking for permission, but keep interruptions to a minimum. During lab, you are expected to keep personal web surfing to a minimum (quickly check your email then get to work). You may not play music or otherwise disturb other students while in the lab. If you finish your lab work early, you may remain in the lab and work quietly. You may be asked to leave after you finish your work if your personal computer usage is disruptive to other students. If you're ahead of the rest of the class and

finish your work early, kindly consider looking around and finding someone who could use your help. You may not invite friends who are not enrolled in the course to join you in the lab to "just hang out".

Plagiarism and Academic Honesty

Academic honesty is expected from all students taking the course. All work that a student submits for course credit must be performed by the student who submits it. Use of for-pay programming services such as RentACoder for programming projects for course credit, either as a buyer or a seller, is not permitted, and is considered a serious offense. Exams must be taken according to parameters established by the instructor at the start of the exam. In general terms, students are prohibited from all communication, both verbal and electronic, with anyone except the instructor during an exam. Notes and other reference aids may or may not be permitted, instructor will specify what resources are allowed, if any, when exam is announced. For programming projects, you may discuss coding details with other students, but you cannot copy and paste large chunks of code from one student's work and then submit it as your own. Violations of this policy can result in a failing grade for the project or exam in question, and depending on the severity of the violation, may result in a failing grade for the course, as well as the violation being reported to the Dean of Academic Affairs. Students who repeatedly violate the policy across multiple courses may be suspended and even expelled from the University.