

SYLLABUS

Spring, 2018

CSC 115 Software Design and Programming II (formerly CS 202J)

4 credits, DII

Prerequisites: CSC110 or CSC 201J

Instructor: Beifang Yi
email: byi@salemstate.edu

Office: MH 211A
Hours: 12:15-2:15pm (TR)
9:30am-10:30am (W)

Phone: (978) 542-7246
Web Site: <http://cs.salemstate.edu/~byi/>

Section	Time	Room	Final Exam
02	W & F 10:50am—12:05pm	MH 206	5/11/Friday 8:00am-10:00am MH 206
L22	W & F 12:15pm—1:30pm	MH 202	

Catalog description:

This course extends the treatment of object-oriented methodologies, languages and tools begun in CSC201J. The emphasis is on the analysis of complex problems, particularly those involving multiple design alternatives, and the use of class libraries. Specific topics include inheritance, polymorphism, recursion, stream and file I/O, exceptions, and graphical interface programming. Style, documentation, solution robustness, and conformance with specifications are emphasized throughout. Three lecture hours and three hours of scheduled laboratory per week plus extensive programming work outside of class.

Prerequisite: CSC110 or 201J.

Course Goals:

The purpose of this course is to enhance and extend students' understanding of tools and techniques for object-oriented software development. Upon completion of the course, a student should be able to do the following:

- CG1: analyze a problem statement for completeness and clarity;
- CG2: use the methodology of object-oriented design to develop class diagrams (data descriptions and methods) for a problem solution;
- CG3: convert this solution into source code in the designated high-level programming language in accordance with a well-defined set of standards;
- CG4: debug and test the program;
- CG5: provide clear documentation for the result.

Course Objectives:

- CO01: gained a deeper understanding of object-oriented design methodology;
- CO02: learned to recognize situations in which multiple design alternatives are possible;
- CO03: learned to recognize and apply design patterns;
- CO04: learned and utilized techniques for validation and verification of programs;
- CO05: gained experience in judging the effectiveness and cost of a software design;
- CO06: gained experience in choosing among competing design alternatives;
- CO07: I gained experience in the use of the UML modeling language;

- CO08: extended their knowledge of an object-oriented programming language, including graphical user interfaces, event-driven programs, file-based input/output, and the use of libraries;
- CO09: produced full documentation for multiple completed projects, including formal class diagrams;
- CO10: participated in one or more group projects.

Course Topics:

A detailed topics list and a general course bibliography can be found on the Computer Science Department website at <http://cs.salemstate.edu/dept/index.php?page=184>. Select CSC 115 to access a PDF document.

Text:

(Required) Java How to Program: Early Objects, 11th Edition, by Deitel & Deitel. Prentice-Hall, 2018 (ISBN: 978-0-13-474335-6).

Required Material:

(Required) Thumb (flash) drive, 4 GB minimum or online storage.

Software:

(Required) J2SE 8.0 (or above) and NetBeans 8.0 (or above) (this is the only IDE that will be covered in class). Free copies of the software that have been customized for the course can be downloaded in the Department labs - instructions will be given in class.

Cell phones:

Turn the ringer off, or, better yet, *turn the phone off*.

Lecture/Lab Attendance:

Class policy is that of the Registrar's office- see the University catalog for details. **Note** that *you are at all times responsible for materials and assignments discussed in class*. We will use SSU's online course management system, Canvas (<https://elearning.salemstate.edu/>) to post assignments, test grades, and announcements regarding the course topics and progress. You will need to visit Canvas (with your SSU Navigator use-name and password) for the course activities. Canvas uses your **SSU-stored email** for the communication between you and the instructor and thus you **must use this email address**. Each student is responsible for completing all course requirements and for keeping up with all that goes on in the course (whether or not the student is present).

Class attendance is strongly recommended. Some of the class hours will be used to review or present software tools, to discuss and investigate Java implementation details that time may not permit to be fully presented in the texts (for design and implementation drills, for programming exercises,) to assist with design and debugging problems that arise in longer lab / project exercises, to discuss and review important topics/questions regarding the tests, and to check/examine/grade the exercises and homework.

Student-Instructor Communication:

Learning how to develop software is very much a **hands-on, experiential process** - the only way to be sure that you understand the material is to apply it by designing and writing programs. The nature of programming is such that it is relatively easy to "get stuck" on minor technical topics that can be difficult to recognize, particularly at early stages of this course - this can lead to a significant amount of what feels like wasted time.

If you have any questions regarding course material, and *in particular if you are having problems with a programming assignment*, the most effective way to get assistance is to *discuss with the instructor (either in the class or outside the classroom)*.

Final Grade:

Final grade will be determined using the following grading weights and formula:

	Grade-A	Grade-B
written assignments	10%	0%
programming assignments	45%	45%
midterm exam	15%	15%
final examination	30%	40%
final grade (overall)	Max(Grade-A, Grade-B)	

Two different grading formulae are used to calculate your semester overall final grade: *Grade-A* and *Grade-B* and your final grade will be the larger of them. Note: it is **more difficult** to get same grade through **Grade-B** than through Grade-A; it is easier to get higher grade through Grade-A scheme.

Attendance is not used to calculate the final grade: however, note that you are at all times responsible for assignments and materials presented in class.

The following table shows how the course work is assessed against the course objectives:

	Quizzes	Assignments (including programming projects and lab exercises)	Exams
CO01	✓	✓	✓
CO02	✓	✓	✓
CO03	✓	✓	✓
CO04	✓	✓	✓
CO05	✓	✓	✓
CO06	✓	✓	✓
CO07	✓	✓	✓
CO08	✓	✓	✓
CO09		✓	
CO10		✓	

Written Assignments:

The questions of written assignments are designed to help understand the course topics, prepare for programming practices, and get ready for tests. And **more important**: many of test questions in both midterm and final examinations will be much like the some of the written assignment questions!

While the solutions for most of the written assignment questions are provided, you will need to present your answers—handwritten or typed. No screenshots of the provided solutions or answers in copy-paste formats will be accepted. You show your solutions to the instructor and **sign your names or initials** on the assignment-checking/grading forms by the deadlines.

Please note: to get full credits for each assignment, **you must sign your names/initials on the assignment-grading sheet** provided by the instructor. Should something unexpected or emergent happened that prevents you from showing your solutions and signing your names/initials on the grading-sheet on time, this is what is called “documented/emergency situation” –please see “Documented Emergency Situation” under Due Dates/Times below. Also note that the instructor will post the assignment grades regularly (usually within one week after the assignment’s due date) on Canvas and you are encouraged to check your assignment grades regularly.

Readings will be assigned from the text on a regular basis: for the maximum benefit from reading, do the readings before the material is covered in class, and you will find that some of written assignment questions are based on the reading of the text.

Programming Assignments:

A significant number of programming assignments/projects will be assigned during the semester. *Most of them will have pre-lab activities to be completed prior to the implementation of the assigned tasks.* They will definitely require **significant programming time outside of scheduled lab.**

Each programming assignment must be submitted at Canvas (<https://elearning.salemstate.edu/>) and **must be tested/examined/checked in the lab by the instructor for full credits** of the assignment.

Programming assignments may have different full score points, depending on the difficulty and the amount of the work of the exercises. There will be one or more programming assignments/projects given in the semester and these extra assignments will be used as make-up assignments. Any assignment may be used as the extras/make-ups. The average score for the overall assignments

will be the total scores received for all the assignments divided by the total scores of the required assignments. For example, suppose that there will be about 1100-point assignment questions given in the semester and that the required assignment total scores will be about 1000 points. You may not move the extra points to the final grade. For example, students A and B have completed 1050-point and 800-point assignments and their Semester Assignment Grades will be $1000/1000 * 45 = 45$ and $800/1000 * 45 = 36$ points respectively (suppose that the required assignment points is 1000).

Please note: to get *full credits* for each assignment, **you must (0) submit your solution to Canvas by the deadline, (1) show your solution in the lab, (2) have the instructor test your code in the lab, (3) answer the questions provided by the instructor correctly during the test, (4) modify/update your code correctly as a solution for a programming question that is very similar to the one you have just submitted/shown, (5) and sign your names/initials on the assignment-grading sheet** provided by the instructor. Should something unexpected or emergent happened that prevents you from showing your solutions and signing your names/initials on the grading-sheet on time, this is what is called “documented/emergency situation” –please see “Documented Emergency Situation” under Due Dates/Times below. Also note that the instructor will post the assignment grades regularly (usually within one week after the assignment’s due date) on Canvas and you are encouraged to check your assignment grades regularly.

- Given the lengthy steps of testing your solution code in the lab, please do submit your answers by deadline and be sure that you are very familiar with your solution—the Java code of *your own work!*
- If during the test in the lab you cannot answer the provided question correctly or make necessary changes upon your submission for a varied problem (which is very similar to the assignment problem), your scores for that assignment will be severely affected (which can be as low as **0%**).

Exams:

There will one midterm exam and one *comprehensive* final examination.

Please refer to Final Grade above for the grading weights of the midterm and final examinations.

Missed Tests:

Tests (exams) may not be made up except for *documented/emergency* situations. If a test must be made up, arrangements must be made with the instructor to take the test before it is discussed in class (usually within a week of the test being administered).

Due Dates/Time:

- **There will be penalties for late submissions: 50% penalty for those being 1 second late up to 1 week; 100% for 1 or more weeks late.**
 - **To make this clearer:** if the submission is one second late, 50% of the credits will be taken off; if one full week or more late, you will get zero (i.e., 0) for that assignment.
 - **Please double-check** your submissions: since your assignment submissions are usually graded after their deadlines so when your submission for an assignment is “incomplete”, the highest grade for your resubmission (of a complete and correct solution) is 50% of the full credits for that assignment.
 - **Please sign on the assignment-grading sheet** once the instructor has checked/examined/graded your assignments in the class/lab.
- **Documented/Emergency Situations:** if something unexpected/emergent happened that prevented you from submitting your assignments on time, you would need to contact the instructor ASAP through email with a certain “proof” (usually some type of documents), and the instructor will reply with a new deadline for that assignment. You will need to save these two emails.
- **No assignments will be accepted after the final examination.**

Study Groups:

While I strongly encourage study groups, I require that each student hand in his/her answers in her/his own words - if two answers are highly similar to each other, neither will receive credit.

When working on your programming projects, you may discuss with others the project topics, the algorithms and methodologies related to the project; but when you work on writing the code, this coding work should be 100% of your own work. **If two answers/written code segments come out exactly the same or highly similar, neither will receive credit and/or further actions will be taken** (such as reporting to the department and/or university). Given the nature of most of the projects, homework questions and writing assignments, it will be almost impossible for two people to come up with highly similar answers UNLESS they copy.

Academic Integrity:

Academic Integrity Policy and Regulations can be found in the University Catalog and on the University's website (http://catalog.salemstate.edu/content.php?catoid=13&navoid=1295#Academic_Integrity). The formal regulations are extensive and

detailed - familiarize yourself with them if you have not previously done so. A concise summary of and direct quote from the regulations: "Materials (written or otherwise) submitted to fulfill academic requirements must represent a student's own efforts". *Submission of other's work as one's own without proper attribution is in direct violation of the University's Policy* and will be dealt with according to the University's formal Procedures.

All students are expected to be familiar with the academic regulations, including those regarding Academic Integrity, for Salem State University as published in the college catalog. In addition, each student is responsible for completing all course requirements and for keeping up with all that goes on in the course (whether or not the student is present).

Salem State University is committed to providing equal access to the educational experience for all students in compliance with Section 504 of The Rehabilitation Act and The Americans with Disabilities Act and to providing all reasonable academic accommodations, aids and adjustments. Any student who has a documented disability requiring an accommodation, aid or adjustment should speak with the instructor immediately. Students with Disabilities who have not previously done so should provide documentation to and schedule an appointment with the Office for Students with Disabilities and obtain appropriate services

In the event of a University declared critical emergency, Salem State University reserves the right to alter this course plan. Students should refer to saalemstate.edu for further information and updates. The course attendance policy stays in effect until there is a university declared critical emergency. In the event of an emergency, please refer to the alternative educational plans for this course located at Canvas (<https://elearning.salemstate.edu/>). Students should review the plans and gather all required materials before an emergency is declared.

Please remember that if, for any reason, you decide to drop this course, you **MUST** do so officially through the Registrar's office. The last day to withdraw from a course this semester is **April 13th**.

Note: This syllabus represents the intended structure of the course for the semester. If changes are necessary, students will be notified in writing and via all regular class communication mechanisms (class discussion, emails, and/or the course link at Canvas <https://elearning.salemstate.edu/>).