Salem STATE | UNIVERSITY

Computer Science Department
cs.salemstate.edu

SYLLABUS **Spring, 2013**

CSC 202J Software Design and Programming II **4 credits**
Prerequisites: CSC 201J with a grade of C+ or higher

Instructor: Beifang Yi
email: byi@salemstate.edu
Office: MH 211A
Hours: TWRF (11:00am-1:00pm)
Phone: (978) 542-7246
Web Site: http://cs.salemstate.edu/~byi/

Sectio n	Time	Room	Final Exam
S1	Th 6:00—10:00pm	MH 202	**Thursday 5/2, 8:00pm-10:00pm 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: CSC201J with a grade of C+ or higher.

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 o 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: l 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 202J to access a PDF document.

Text:

(Required) Java How to Program: Early Objects, 9th Edition, by Deitel & Deitel. Prentice-Hall, 2012 (ISBN: 0132940949).

Required Material:

(Required) Thumb (flash) drive, 2 GB minimum

Software:

(Required) J2SE 6.0 (or 7.0 or above) and NetBeans 6.8 (7.0 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.

Cell phones:

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

Lecture Attendance:

Class policy is that of the Registrar's office - see the University catalog for details. Lecture will start promptly at the scheduled time, so please make a serious effort to not be late; if you *have* to be late, please be discrete when entering the classroom. While class attendance does not *directly* affect your final grade, some of the material covered in class is not found (in the same form) in the text, so class attendance and notes are very important. Note that you are at all times responsible for materials and assignments discussed in class: if you miss a class, try to get lecture notes from a classmate and review them *before* the next lecture. We will use SSU's online course management system, Canvas (https://salemstate.instructure.com/login) to post assignments, quiz 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*** box 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).

The once-a-week class meeting will start with a quick review of the topics covered in the previous week and the instructor will answer any questions about them. Then new topics will be introduced and explained in detail through lecturing and lab demonstrations and practices. The last session of the meeting will present new assignments (and explanations), which will be followed with quizzes (in some weeks). **Class attendance (particularly during *lab sessions*) is *strongly* recommended**. Lab will be used to review or present software tools, to discuss and investigate Java implementation details that time may not permit to be fully explored during the scheduled lecture period, for design and implementation drills, for occasional short lab (programming) exercises, to assist with design and debugging problems that arise in longer lab / project exercises, 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. While the single most effective way to deal with these problems is to talk to the course instructor, that approach can be problematic if the class meets only once or twice a week and/or if the instructor's office hours conflict with students' obligations.

If you have any questions regarding course material, and *in particular if you are having problems with a programming project*, 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:

assignments (labs/project or short-answer exercises)	30%
quizzes	30%
midterm examination	13%
final examination	27%

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		✓	

Assignments (Laboratory / Projects/Short-Answer Exercises):

10-15 exercises (including short-answer exercises, programming projects) will be assigned during the semester. *Most will have pre-lab activities to be completed prior to the implementation of the assigned tasks while a few will be in the form of short answer*. Exercises *will definitely require* ***significant*** *programming time outside of scheduled lab*. Submission requirements and mechanics will be stated on each exercise In general, each exercise will have an assigned due date and time: the required material(s) are to be submitted no later than midnight of that date. Please refer to Final Grade above for the grading weight of the assignments.

*Each assignment may have different full score points, depending on the difficulty and the amount of the work of the exercises. There will be one or two extra assignments given in the semester and these extra assignments will be used as make-up assignments. Any assignment may be used as the extras. 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. Usually there will be about 1100-point assignment questions given in the whole semester and 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.*

There will be ***Challenging Programming Projects****, which will be graded separately.* ***No*** *late submission of these projects will be accepted.* ***Grades earned from these Challenging Projects will be added to the final grade****. For example, if student C has completed 5 point Challenging Projects, this 5 points will be added to his/her final grade calculated from the table illustrated in Final Grade above.*

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. Supplementary material will be distributed on a regular basis, *and will be the primary focus of class discussions.* In particular, at least one complete, functioning example Java application project will be distributed per week or biweekly: these projects include extensive student-oriented documentation/comments that are designed to guide students through the process of learning how to design and implement software. Occasional worksheets and problems will be assigned.

Exams/Quizzes:

There will be two exams, a midterm (usually in week 8) examination and a *comprehensive* final examination.

There will be about 10 quizzes to be held in class or lab hours (one or two quizzes with the lowest grades will not be used to calculate the final grading/scoring). There are different forms of quizzes: *short-answer questions* (paper-based or online, including coding practice questions) and complete *Java Programming* Quizzes.

The **Java Programming Quiz**: *This form of quiz is in the format of Java coding project and will be held in the lab and completed on the lab machines. You may use any paper materials (i.e., open book, open notes) but you* ***may not, must not*** *use the Internet or any documents/data in electronic formats (such as code/data on the hard drive, CD, USB, etc) for the quiz.*

Java Programming Quiz Grading: we will use a programming judging system to automatically grade your quiz. To get a full score for a quiz, (1) Your submitted code must pass through the grading/examination of the judging system (which mentions No Compilation Error, **and** 100% correct results in the execution); (2) You code must also meet the quiz requirements. This means that if your code does not pass through the auto-grading (even if you have completed 99% code), you quiz grade will be 0 and that the successful pass through the auto-grading (no errors and correct result output) does not guarantee a full score (if your code fails to meet some requirements, for example, submit one Java class when two are required).

Please refer to Final Grade above for the grading weights of the exams and quizzes.

Missed Tests:

Tests (exams and quizzes) 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 a 5% penalty for each week an assignment (lab/project/short-answer exercise) is late**; penalties accrue at the due time of the assigned due date.
- **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.

"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 http://www.salemstate.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 http://cs.salemstate.edu/~byi/2013Spring/CSC202J/emergency/index.html . 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 12th**.

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 instructor's website at http://cs.salemstate.edu/~byi/).