**SYLLABUS**                                                                                   **Fall, 2013**

**CSC 260 Data Structures and Algorithms**                                              **4 cr.**
**Prerequisites: CSC 200A and CSC 202J**

**Instructor**:  Beifang Yi                        **Office**: MH 211A              **Phone**: (978) 542-7246
**email**:        byi@salemstate.edu          **Hours**: W & F (9:00-10:40am)    **Web Site**: http://cs.salemstate.edu/~byi/
                                             T (1:30-3:30pm), R (4:00~6:00 & 10:00pm~10:30pm)

| Section | Time | Room | Final Exam |
|---|---|---|---|
| **S1** | R 6:00pm-10:00pm | MH 202 | **Thursday, 12/19, 8:00pm-10:00pm** <br> **MH 202** |

**Catalog description:**

  Basic data structures such as stacks, queues, linked lists, and trees are studied and applied to problems in data storage and manipulation.  Applications include basic searching and sorting algorithms.  Design, analysis and implementation techniques are discussed.  Three lecture hours and three hours of scheduled laboratory per week, plus extensive programming work outside of class.

**Goals:**

  The purpose of this course is to develop students' knowledge and appreciation of organization and retrieval techniques and to familiarize students with the basic concepts of order-of-magnitude analysis.  The goals of this course are:

  CG01:    to develop an appreciation for the process of data abstraction and its usefulness in software development;
  CG02:    to develop the skills and knowledge necessary to perform basic analysis of algorithms;
  CG03:    to present a selection of the most common data structures and their standard implementations and uses;
  CG04:    to present a selection of the most common algorithms for searching and sorting.

**Objectives:**

  Upon successful completion of the course, student will have:

  CO01:    applied data abstraction techniques;
  CO02:    implemented several classic data structures "from scratch";
  CO03:    demonstrated knowledge and use of ADTs available in one or more language libraries;
  CO04:    recognized the factors required to perform analysis of algorithms and performed order-of-magnitude analysis;
  CO05:    chosen, with justification, an appropriate structure to match the requirements of a given problem, implemented the structure if necessary, and used it in an appropriate way to solve the problem;
  CO06:    utilized standard techniques for program validation;
  CO07:    demonstrated the ability to use the UML modeling language;
  CO08:    produced documentation for at least one major completed project, including formal class diagrams and rigorous test set specification and results.
  CO09:    participated in at least one group project involving problem analysis and design specification and selection
  CO10:    demonstrated recognition of the need for future professional development through research into future trends in the areas of analysis of algorithms  and application development and profiling.

**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 260 to access a PDF document.

**Text:**
    **(Required) Data Structures and the Java Collections Framework**, 3$^{rd}$ Edition, by William J. Collins. Wiley, 2011 (ISBN: 978-0-470-48267-4).

**Required Material:**
    **(Required)** Thumb (flash) drive, 4 GB minimum or online storage (for saving your projects)

**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 - instructions will be given in class.

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

**Class 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).
    Quizzes will be held in class lecture and lab hours.

**Scheduled Lab Attendance:**
    Attendance during lab time is *STRONGLY* **recommended**. Labs 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.

**Final Grade:**
    Final grade will be determined using the following grading weights:

| | |
|---|---|
| assignments (labs/project or short-answer exercises) | 35% |
| quizzes | 25% |
| 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/Exams | Assignments (labs, programming projects, and other exercises) |
|---|---|---|
| CO01 | ✔ | ✔ |

| | | |
|---|:---:|:---:|
| CO02 | ✔ | ✔ |
| CO03 | ✔ | ✔ |
| CO04 | ✔ | ✔ |
| CO05 | ✔ | ✔ |
| CO06 | | ✔ |
| CO07 | ✔ | ✔ |
| CO08 | ✔ | ✔ |
| CO09 | | ✔ |
| CO10 | | ✔ |

**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).*

**Assignments (Lab Exercises/Programming 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 Please refer to Final Grade above for the grading weight of the assignments.

There is a deadline to each assignment and ***penalty*** will be imposed for late submissions (see the Due Dates/Time below).

All the assignments must be submitted at Canvas (https://salemstate.instructure.com/) unless otherwise noted.

Each assignment may have different full score points, depending on the difficulty and the amount of the work of the exercises. There might be one or two extra assignments given in the semester and these extra assignments will be used as make-up assignments. All the assignment (including the extra assignments) questions might be in the examination. Any assignment may be used as the extras. 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. You may not move the extra points to the final grade. For example, if there are about 1100-point assignment questions throughout the semester and the required assignment total scores will be about 1000 points, and if students A and B have completed 1050-point and 800-point assignments, their overall semester assignment (written) grades will be 100% and 80% respectively for A and B).

*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 directly added to the final grade**. For example, if student C has completed 5 point Challenging Projects, these 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. Check the above for examination grading weights and times.

Quizzes will be held in class/lecture, lab, and outside-class/lab (i.e., take-home) hours. There will be about 6 to 9 quizzes to be held in the semester (one quiz 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* (in-class or take-home) Quizzes.

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.
  - Though technically you may still get credits for your late submissions, this will severely affect your course work ahead and particularly your final project. **Note again**: this course covers *so many* different topics with *various* implementation techniques. Your timely completion of the assignments is the most important factor for your success in this course.
- **No assignments (lab/project/short-answer exercise…) will be accepted after the final examination**.

**Study Groups:**

While I strongly encourage study groups, for non-group assignments I require that each student hand in his/her answers in his/her own words **- if two answers 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/2013Fall/CSC260/emergency/index.html. Students should review the plans and gather all required materials when 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 **November 22nd**, **Friday**.

> **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 homepage through the instructor's website at http://cs.salemstate.edu/~byi/).