Salem STATE | UNIVERSITY

Computer Science Department
cs.salemstate.edu

SYLLABUS **Spring, 2025**

CSC 115 Software Design and Programming II **4 cr. DII**

Instructor: Beifang Yi **Office**: MH 208B **Phone**: (978) 542-7246
email: byi@salemstate.edu **Hours**: W-F (1:30-4:00pm) **Website**: http://weblab.salemstate.edu/~byi/

Section	Time	Room	Final Exam
01	W & F 10:50am-12:05pm	MH 202	**May 8, Thursday 11:00am-1:00pm MH 202**
L21	W & F 12:15-1:30pm	MH 202	
Office Hours (MH208B)		Wednesday & Friday (1:30-4:00pm)	

Catalog description:
This course extends the treatment of object-oriented methodologies, languages and tools begun in CSC110. The emphasis is on the analysis of complex problems, particularly those involving multiple design alternatives, and the use of class libraries. Fundamental strategies for algorithm design are presented and discussed. 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.

Prerequisites: CSC110 or ITE210.

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:

- CG01: analyze a problem statement for completeness and clarity;
- CG02: use the methodology of object-oriented design to develop class diagrams (data descriptions and methods) for a problem solution;
- CG03: demonstrate understanding of and apply fundamental strategies for algorithm design;
- CG04: convert this solution into source code in the designated high-level programming language in accordance with a well-defined set of style rules;
- CG05: debug and test the program;
- CG06: provide clear documentation for the result.

Course Objectives:
By the end of the course students will have:

- CO01: gained a deeper understanding of object-oriented design methodology;
- CO02: learned to recognize situations in which multiple design alternatives are possible;
- CO03: applied fundamental algorithm design strategies;
- CO04: learned to recognize and apply design patterns;
- CO05: learned and utilized techniques for validation and verification of programs;
- CO06: gained experience in judging the effectiveness and cost of a software design;
- CO07: gained experience in choosing among competing design alternatives;
- CO08: gained experience in the use of the UML modeling language;
- CO09: 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;

- CO10: produced full documentation for multiple completed projects, including formal class diagrams;
- CO11: 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/courses/course-information-documents (or at http://cs.salemstate.edu/courses and then select "Degrees & Courses") and select "CSC 115 Software Design and Programming II" to access a PDF document.

Text:

(Required) Java How to Program: Early Objects, 11th Edition, by Deitel & Deitel. Prentice-Hall, 2017 (ISBN: 978-0-13-474335-6).

Course Materials & Software:

Thumb (flash) drive, 64 GB minimum or online storage (for saving your projects and coursework) in addition to your personal computer/laptop (Windows, MacBook, or Linux machines).

Download the JDK (i.e., Java Development Kit) at https://www.oracle.com/java/technologies/downloads/. Get JDK17 or above and install it on your computer. You may use your preferred IDE for working on Java programming assignments, but the IDE is ***not*** required.

You will need to use Microsoft Word or similar software packages to complete some assignments.

It is expected that you work on your laptop for the lab/coding exercises during the lab hours. If you need technical help regarding your computer configuration or setup issues including software installation, please contact Information Technology Service (ITS) (at https://www.salemstate.edu/offices-and-services/information-technology-services).

Additional references:

- Course teaching materials: http://weblab.salemstate.edu/~byi/CSC201J_202J_Deitels/index.html
  - This website is password-protected and ask the instructor for the password (or log into Canvas for it).
- Course online system (Canvas): https://elearning.salemstate.edu/.
  - Access to this site via the username and password given/assigned by SSU.

Class/Lab Attendance:

Regular attendance in both class and lab sessions contributes significantly to your coursework and particularly to your final grade. Lab exercises will be evaluated and graded during designated lab periods, with ***no*** exceptions for late submissions. Certain tests will be administered during class or lab hours.

Class and lab time will be allocated for a variety of activities, including detailed explanation of the course topics, comprehensive review of course material, in-depth exploration of Java implementation details beyond textbook coverage, practical application exercises, troubleshooting project-related issues, test preparation and administration, and assessment of assignments and homework.

Lectures will commence promptly at the scheduled time, and students are expected to arrive on time. All course content, including assignments, grades, and announcements, will be accessible through Salem State University's online course management system, Canvas (https://elearning.salemstate.edu/). Students must use their **SSU Navigator credentials** to access Canvas and ensure their SSU email address is current for communication with the instructor.

It is the student's responsibility to complete all course requirements and stay informed about course content, regardless of attendance.

Grading Policies & Course Assessments:

Final grade will be determined using the following grading weights:

reading/writing assignments	5%
lab exercises	30%
programming projects	20%
mini-tests (quizzes)	20%
final examination	25%

Although attendance is not factored into the final grade, lab and project testing and grading will take place during lab hours. Additionally, mini-tests (quizzes) will be scheduled during class or lab hours. You are always responsible for completing all assignments and for understanding the materials presented in class.

The numeric final grade will be converted to a letter grade based on the following grading system and this letter grade will be submitted as the official grade for the course.

Overall Final	**Letter Grade**
94-100	A
90-93	A-
87-89	B+
84-86	B
80-83	B-
77-79	C+
74-76	C
70-73	C-
67-69	D+
64-66	D
60-63	D-
0-59	F

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

	Test/quiz Questions	Homework Problems	Programming Projects	Lab Exercises
CO01	✓	✓	✓	✓
CO02	✓	✓	✓	✓
CO03	✓	✓	✓	
CO04	✓	✓	✓	✓
CO05	✓	✓	✓	✓
CO06	✓	✓	✓	✓
CO07		✓	✓	✓
CO08	✓	✓	✓	✓
CO09			✓	
CO010			✓	
CO11			✓	

AI Policy and Coding Assignment Grading:

Online sources and generative AI tools (e.g., ChatGPT) may be used to supplement your study of course topics and concepts. However, the primary approach to learning programming should involve reading the assigned textbook, engaging with class lectures, completing lab exercises during lab hours, and following the provided examples when working on coding assignments. Online sources and AI tools should **only serve as supplementary aids**.

Submitting solutions (i.e., programs) found online or generated by AI tools as your own work for coding assignments (e.g., labs, projects, etc.) is considered plagiarism**.**

Please be aware that online "solutions," especially those generated by AI tools, may incorporate "advanced programming techniques" that have not been introduced in the course by the assignment deadline. You are ***only*** permitted to use programming techniques that have been taught up to that point. Failure to adhere to this requirement may result in significant grade reductions or a grade of zero for coding assignments.

Tests (mini-tests/quizzes and final examination):

Mini-tests (quizzes) will be administered throughout the semester, along with a comprehensive final exam. The weight of each assessment in determining your final grade is outlined in the grading policies above.

Mini-tests will consist of closed-book (and/or online) questions in multiple-choice, true/false, fill-in-the-blank, and short-answer formats, as well as programming problems. These tests must be completed during designated lab periods. Programming mini-tests will resemble smaller-scale versions of lab exercises and projects.

The final exam will be a comprehensive evaluation of all course material.

Missed Tests:

Make-up exams (final exam or mini-tests) are generally ***not*** permitted unless there is documented proof of an emergency. If you need to reschedule a mini-test, you must arrange this with the instructor within one week of the original test date. The final exam make-up will be done in the university designated Make-up Exam Period.

Lab Exercises:

Lab exercises are designed not only to help you understand the course topics but also to prepare you for the programming projects. Lab exercises must be completed, tested, and graded by the instructor during lab hours. Submitting your lab work to Canvas by the deadline alone does ***not*** guarantee credit—you must have it tested and answer questions given by the instructor to receive a grade. Details can be found in Lab Syllabus document.

Lab exercises must be submitted on time by the specified deadlines; late submissions will receive a grade of **zero**.

Please note that lab exercises and programming projects are separate assignments, and the credits earned from programming projects do not count toward your lab assignment grades, even though some labs may be part of projects.

Some tests will also be conducted during lab hours.

The table below outlines the lab activities (subject to minor adjustments as we progress through the coursework.

Week	**Dates**	**Contents (textbook chapters and others)**
1	1/13—1/17	Reviews/Programming Basics (Ch 1, 2, 3)
2	1/20—1/24	Design & Implementation Basics (Ch 4, 5)
3	1/27—1/31	Design & Implementation (Ch 6, 7)
4	2/3—2/7	Design & Implementation (Ch 7, 8)
5	2/10—2/14	OOP: Inheritance (Ch 9)
6	2/17—2/21	OOP: Inheritance (Ch 9) OOP: Polymorphism & Interface (Ch 10)
7	2/24—2/28	OOP: Polymorphism & Interface (Ch 10)
8	3/3—3/7	Exceptions & Exception Handling (Ch 11 & others)
9	3/10—3/14	Java GUI (Ch 2, 3, …, 11)
10	3/17—3/21	**Spring Recess**
11	3/24—3/28	Java GUI (Ch 2, 3, …, 11)
12	3/31—4/4	Strings, Characters (Ch14)
13	4/7—4/11	Files, I/O Streams (Ch15) Generic Collections (Ch16)
14	4/14—4/18	Lambdas and Streams (Ch17) Recursion (Ch18)
15	4/21—4/25	Searching, Sorting (Ch19) Generic classes/Methods: A Deeper Look (Ch20)
16	4/28—5/1	**Reviews**

Lab Exercise Submission:

For each lab exercise, **a lab report** is required for submission in the format of Word or PDF document, the content of which depends on the lab exercise—check the lab requirement description for detail. In addition, the report must contain the following:

- A short paragraph (several sentences about 30~150 words in length) of short description of the lab (***what*** about the lab and ***how*** to solve the problem);
- A screenshot of the computer system time when you have completed the lab—this timestamp must come from the computer system you have used for the completion of the lab.

Most of the labs are coding related exercises; in such cases, you must submit the following **additional items**:

- In the same report, screenshots of the executions of the lab programs (i.e. the running of Java programs)—the execution process may consist of many inputs and outputs. The screenshots do not have to cover all the inputs/outputs but should include part of the them.
- Java program files (i.e., **.java** files): upload all Java files—do not copy these Java programs to the report document—upload the **.java** files.

Do ***not*** zip or compress the submission files.

Ensure that the contents of your screenshots are **clear and easy to read**. Avoid capturing entire monitor screens or full IDE windows; instead, focus on the specific required content. If the screenshots are not legible, you may receive *zero* points for the lab.

Lab Grading Procedures:

1. Submit your solution (Java source code, lab report) for the lab assignment to Canvas by the deadline;
2. Show/test your solution in the lab hours;
3. Have the instructor test your code in the lab hours;
4. Answer the questions provided by the instructor correctly during the testing.

Due Dates/Time:

- Lab exercises must be submitted on time by the specified deadlines; late submissions will receive a grade of zero.
- For **all other coursework**, late submissions will incur the following penalties:
  - Up to 24 hours late: A deduction of 25 percentage points.
  - More than 24 hours and up to 48 hours late: A deduction of 50 percentage points.
  - More than 48 hours and up to 72 hours late: A deduction of 75 percentage points.
  - More than 72 hours late: A grade of zero will be assigned.
  - After the final examination: No assignments, labs, or projects will be accepted after the final exam, and a grade of zero will be given for any submissions made after this time.
- In case of an emergency that prevents you from completing or submitting your assignments/projects on time, you must email a request for an extension. The instructor will respond with a specific extension period or a new deadline for the assignment. Be sure to keep this email as proof of the extension approval. Note that sending the request email alone does **not** guarantee approval.
- Always **double-check your submissions**, as assignments are typically graded after the deadline. To ensure successful submission, download your submitted files and review them carefully. For projects and lab exercises, verify the downloaded programs by compiling and running the code to confirm everything is in order.

Study Groups:

While I strongly encourage study groups, each student must submit their answers in their own words or solutions. If two submissions are highly similar, neither will receive credit.

When working on your programming projects, you may discuss project topics, algorithms, and methodologies with others. However, the coding must be entirely your own work. If two code submissions are identical or very similar, neither will receive credit, and further action may be taken, such as reporting the incident to the department or university.

Collaboration is encouraged for discussing project topics, algorithms, and methodologies. However, all code must be your original work. Identical or highly similar code submissions will result in ***zero*** credit for both parties and may lead to further disciplinary action.

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 University has established comprehensive regulations governing academic integrity. Please familiarize yourself with these guidelines if you haven't already. A concise summary and direct quote from the regulations states: "Materials (written or otherwise) submitted to fulfill academic requirements must represent a student's own efforts." Submitting someone else's work as your own without proper attribution is a direct violation of the University's policy and will be addressed according to the University's formal procedures.

Equal Access Statement:

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 Disability Services and obtain appropriate services.

University Emergency Statement:

In the event of a university declared critical emergency, Salem State University reserves the right to alter this course plan. Students should refer to Salem State 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.

Coursework Expectations and Schedule:

Students enrolled in this four-credit course should plan to spend approximately three hours per week attending class and three hours in lab attendance. Additionally, **a minimum of eight hours per week outside of class and lab** is required for course-related work, beyond the six hours spent in class and lab sessions.

Students are responsible for adhering to Salem State University's academic regulations, including those pertaining to academic integrity, as outlined in the college catalog. It is essential that students complete all course requirements and keep up with course content, even in absences. The following table outlines the course schedule, including the topics covered each week of the semester and the final examination time.

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 4th, 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 link at Canvas https://elearning.salemstate.edu).