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

Instructor: TBA **Office:** location **Phone:** (978) 542-extension
email: TBA@salemstate.edu **Office Hours:** days and times

Section	Time	Room	Final Exam
nn	days and times	location	date and time
Lnn	days and times	location	

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.

Prerequisite: CSC110 or ITE 210.

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.

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.

Student Outcome (SO) vs. Course Objectives matrix

SO	CO01	CO02	CO03	CO04	CO05	CO06	CO07	CO08	CO09	CO10	CO11
SO-1	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓
SO-2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
SO-3		✓			✓			✓		✓	✓
SO-4	✓	✓		✓	✓			✓	✓		
SO-5										✓	✓
SO-6	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓

Notes:

SO-1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

SO-2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

SO-3: Communicate effectively in a variety of professional contexts.

SO-4: Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

SO-5: Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.

SO-6: Apply computer science theory and software development fundamentals to produce computing-based solutions.

Topics (using Java and UML):

- review of: **SDF2(1, 0, 0), PL1(2, 0, 0), SDF1(1, 0, 0), SE5(1, 0, 0)**
  - basic design concepts
  - Java syntax
  - UML
  - the concept of incremental development
- designing for reuse **SDF1(0.5, 0, 0)**
- discovering and applying design patterns **SDF1(0.5, 0, 0)**
- fundamentals of algorithm design **AL2(2, 0, 0)**
  - strategic approaches
    - Greedy, Divide and Conquer, Dynamic (Feedback, Retrospective)
  - comparison of approaches – advantages and disadvantages
- functional programming **SDF1(1, 0, 0)**
  - Effect-free programming **PL2(1,0,0)**
  - first-class functions (taking, returning, and storing functions) **PL2(3, 0, 0)**
- subclasses and inheritance **SDF1(2, 0, 0), PL1(0, 2, 0)**
- file-based input and output **SDF2(2, 0, 0)**
- exceptions and exception handling **SDF4(2, 0, 0)**
- polymorphism **PL1(0, 1.5, 0), SDF1(1, 0, 0)**
- data organization and retrieval
  - sorting algorithms **SDF1(1, 0, 0), AL3(1, 0, 0)**
  - searching algorithms **SDF1(1, 0, 0), AL3(1, 0, 0)**
  - performance analysis **AL1(2, 0, 0)**
  - testing and validation **SE7(0, 1, 0), SE3(1, 0, 0), SDF4(1, 0, 0)**
- interfaces **SDF1(1, 0, 0), PL1(0, 1.5, 0)**
- recursion **SDF2(2, 0, 0)**
- survey of class libraries **PL1(0, 1, 0)**
- programmer-developed windows and frames **SDF4(3, 0, 0), HCI1(2, 0, 0), HCI2(0, 2, 0)**
- multidimensional arrays **SDF3(2)**

Programming assignments: Approximately six programming assignments are given. One or more of these may be group projects. Each programming assignment involves the design, writing, testing and debugging of a program and the submission of an appropriate laboratory report. Each assignment has a specific due date, with a short grace period during which the assignment may be submitted for reduced credit. When the grace period has expired, the assignment will no longer be accepted.

All programs must be coded in the programming language currently used for instruction in the CSC201J/202J sequence - no exceptions will be allowed. The version of the language being used will be the currently accepted standard version: any extensions or variations in student-owned compilers must be approved in advance by the instructor, who may choose to forbid their use.

Laboratory exercises: There will be short programming exercises to be completed during weekly scheduled laboratory sessions. Each exercise focuses on a specific language feature or programming technique presented in recent lectures. Performance on these exercises will be incorporated into the course grade.

Exams and quizzes: There will be a midterm examination and a comprehensive written two-hour final examination.

Final grades will be determined on the basis of the following approximate weights: examinations - 40%, programming assignments and lab exercises - 60%.

Course Objective / Assessment Mechanism matrix

	Test / Quiz Questions	Homework Problems	Programming Projects	Lab Exercises
CO01	✓	✓	✓	✓
CO02	✓	✓	✓	✓
CO03	✓	✓	✓	
CO04	✓	✓	✓	✓
CO05	✓	✓	✓	✓
CO06	✓	✓	✓	✓
CO07		✓	✓	✓
CO08	✓	✓	✓	✓
CO09			✓	
CO10			✓	
CO11			✓	

Bibliography:

Bloch, Joshua. **Effective Java. Third Edition** Addison-Wesley Professional, 2018.

Deitel, Harvey; Deitel, Paul. **Java How to Program: Early Objects Version. Eleventh Edition.** Pearson, 2017.

Evans, Benjamin J.; Flanagan, David. **Java in a Nutshell: A Desktop Quick Reference**. **Seventh Edition.** O'Reilly, 2019.

Farrell, Joyce. **Java Programming. Ninth Edition.** Cengage Learning, 2018.

Gaddis, Tony. **Starting Out with Java: Early Objects. Sixth Edition.** Pearson, 2017.

Gaddis, Tony. **Starting Out with Java: From Control Structures through Object. Sixth Edition.** Pearson, 2019.

Horstmann, Cay. **Big Java: Early Object**. **Seventh Edition.** John Wiley & Sons, 2019.

Horstmann, Cay S. **Core Java ™, Volume I—Fundamentals. Eleventh Edition**. Prentice Hall, 2018.

Horstmann, Cay S. **Core Java ™, Volume 2—Advanced Features. Eleventh Edition**. Prentice Hall, 2018.

Liang, Y. Daniel. **Introduction to Java Programming and Data Structures, Comprehensive Version. Eleventh Edition.** Pearson, 2017.

Lewis, John; Loftus, William. **Java Software Solutions. Nineth Edition.** Pearson, 2017.

Schildt, Herbert. **Java 7: The Complete Reference**. **Eleventh Edition.** McGraw-Hill, 2018.

Academic Integrity Statement:

"Salem State University assumes that all students come to the University with serious educational intent and expects them to be mature, responsible individuals who will exhibit high standards of honesty and personal conduct in their academic life. All

forms of academic dishonesty are considered to be serious offences against the University community. The University will apply sanctions when student conduct interferes with the University primary responsibility of ensuring its educational objectives." Consult the University catalog for further details on Academic Integrity Regulations and, in particular, the University definition of academic dishonesty.

The Academic Integrity Policy and Regulations can be found in the University Catalog and on the University 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. *Copying without attribution is considered cheating in an academic environment - simply put,* ***do not do it!***

University-Declared Critical Emergency Statement:

In the event of a university-declared emergency, Salem State University reserves the right to alter this course plan. Students should refer to 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, which will be distributed via standing class communication protocols. Students should review the plans and act accordingly. Any required material that may be necessary will have been previously distributed to students electronically or will be made available as needed via email and/or Internet access.

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 the Office for Students with Disabilities and obtain appropriate services."

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 email.