

Note: Given the nature of the relationship between CSC 498 and CSC 500, the course description for each course is presented, followed by the goals and objectives presented as a single integrated unit.

CSC 498 Project Specification and Design Practicum

1 cr.

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

Catalog description:

The practicum sets up a typical environment for the development of a detailed proposal for a software- or hardware-system project. The instructor will assist each student in choosing an appropriate project topic and in refining the proposal through all stages from initial outline to final formal specification. The completed proposal will serve as the contract for the CSC 500 Directed Study project. The course involves periodic meetings, group discussions, and individual conferences. The practicum is graded on a Pass/Fail basis and is taught on a Directed Study basis. Open only to Computer Science majors.

Prerequisites: CSC 300 and permission of Department Chairperson.

CSC 500 Directed Study in Computer Science - I

3 cr.

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

Catalog description:

A substantial project involving system design and implementation is carried out on an individual or group basis under the supervision of a faculty member. The specification for the project must have been completed in the prerequisite course CSC 498. Open only to Computer Science majors.

Prerequisites: CSC 498. Additional prerequisites, which vary with the project, are at the discretion of the faculty supervisor for the project.

CSC 498 Project Specification and Design Practicum / CSC 500 Directed Study in Computer Science I

Goals:

The purpose of this course sequence is to develop students' ability to construct and implement a proposal for a project in Computer Science. The goals of this sequence are:

- CG01: to develop an appreciation for the process of formulating a project for implementation;
- CG02: to develop the skills necessary to assess a project proposal for appropriateness and feasibility;
- CG03: to further develop the skills and knowledge necessary to analyze, design, implement and verify system or software projects;
- CG04: to give students experience in making and critiquing presentations.

Upon completion of the course sequence, a student will have demonstrated the ability to perform the activities and techniques necessary to identify a potential development target, developed a design, selected the tools utilized during implementation, and have implemented, verified and evaluated a solution.

Objectives:

Upon successful completion of the course sequence, students will have:

- CO01: demonstrated knowledge of the initial phases of the project development life cycle;
- CO02: demonstrated knowledge of the major process models used in the development of large-scale systems;
- CO03: demonstrated knowledge of the tools and techniques appropriate for implementation of the project, specifically including UML;
- CO04: demonstrated knowledge of modern design paradigms;
- CO05: developed a plan for project implementation;
- CO06: presented and defended a project proposal and solution design to the Computer Science faculty and students;
- CO07: carried out an implementation plan, recording any deviations from the plan along with rationale and ramifications;
- CO08: presented an analysis of a completed project to the Computer Science faculty and students.

Program Outcome vs. Course Objectives matrix

Program Objective (condensed form)	CO01	CO02	CO03	CO04	CO05	CO06	CO07	CO08
PO-A: apply knowledge of computing and math	✓	✓	✓	✓	✓	✓	✓	✓
PO-B: analyze a problem and define its computing requirements	✓	✓	✓	✓	✓	✓	✓	✓
PO-C: design, implement and evaluate applications				✓			✓	✓
PO-D: function effectively in teams to accomplish a common goal								
PO-E: professional, ethical, and social responsibilities								
PO-F: communicate effectively with a range of audiences						✓		✓
PO-G: local and global impact of computing on people and society								
PO-H: need for continuing professional development		✓	✓	✓			✓	✓
PO-I: use current techniques, skills, and tools		✓	✓	✓	✓	✓	✓	✓
PO-J: apply theory and principles to model and design systems		✓	✓	✓		✓		✓
PO-K: apply design and development principles in constructing software		✓	✓	✓	✓	✓	✓	✓
PO-L: apply knowledge of computing and mathematics appropriate to the discipline	✓	✓	✓	✓	✓	✓	✓	✓
note - full statements of the Program Outcomes (program objectives) for the Computer Science Major can be found in the document <i>Computer Science Major Program Educational Objectives and Program Outcomes</i> on the Assessment page of the Computer Science Major (cs.salemstate.edu)								

Note: All projects are expected and required to relate to the specific Program Objectives as indicated above. In addition, a specific project may relate to Program Objectives PO-D, PO-E, and PO-G depending on the nature of the project.

Topics:

- review of the initial phases of the development of a formal proposal SE5(2)

- investigation of general needs
- analysis of existing functionalities
- proposal of a set of new/modified functionalities
- review of the systems development process **SE4(2), SE1(1)**
 - basic principles
 - the development life cycle
- review, as necessary, of systems analysis techniques **SE1 (1)**
 - information gathering
 - team communication
 - feasibility studies
- review of data analysis and modeling techniques **SE1(1)**
- overview of systems architecture **SE1(1)**
 - for software projects, primary focus on ADTs, object recognition and specification, and file/database design (if appropriate)
 - for hardware projects, primary focus on system block diagrams, system circuit diagrams and wiring diagrams
- review of general implementation issues **SE6(2), SE7(0.5)** SE11(1) (nc)
 - reliability
 - testing
 - verification
 - maintenance (including modifiability)
 - evolution

The primary goal of this course is to guide students through the process of designing a system and specifying its implementation requirements. Students will choose an application arena of sufficient complexity so as to necessitate a non-trivial solution to the problem of designing and implementing a solution for the application. This topic area will then be studied through reading and discussion. After a thorough analysis of the functionalities required by the application area, the students will develop and present to the project supervisor various data modeling and system architecture possibilities: the possibilities will be discussed and evaluated, leading to a final document that:

- describes the functionalities of the proposed system in clear, concise and non-technical terms;
- specifies the tools necessary to implement a solution;
- defines a high-level design architecture for a solution;
- specifies important developer-designed objects required to represent the application area;
- describes the implementation techniques that are appropriate for manipulating the objects;
- presents an implementation schedule;
- presents a mechanism for determination of the final grade for CSC 500.

The finished proposal will be presented to the department Directed Study Committee and to the department at large during the final week of the semester (typically on Reading Day). Final approval of the proposal will be determined by majority vote of those members of the Directed Study Committee who are present at the presentation, said vote taking place immediately after all presentations have been completed.

The finalized document will act as the contract document for the project that is to be implemented in CSC 500. Any changes to the finalized approved document must be approved, in writing, by the student, the supervising faculty member, and the department Directed Study Committee. The (pass/fail) grade for the Practicum will be based on the final document (in particular on the analysis of the required functionalities, the scope of the project, and on the appropriateness of any proposed design(s)) and the quality of the presentation and defense of the proposal.

Course Objective / Assessment Mechanism matrix

	Proposal			
	Problem Specification	Proposed Solution Design	Proposed Implementation Techniques and Tools	Presentation
CO01	✓	✓	✓	✓
CO02	✓	✓	✓	✓
CO03	✓	✓	✓	✓

	Proposal			
	Problem Specification	Proposed Solution Design	Proposed Implementation Techniques and Tools	Presentation
CO04	✓	✓	✓	✓
CO05	✓	✓	✓	✓
CO06	✓	✓	✓	✓
CO07	✓	✓	✓	✓
CO08				

	Completed Project				
	Specification Component(s)	Implementation Component(s)	Result Analysis Component(s)	Documentation Component(s)	Presentation
CO01	✓	✓	✓	✓	✓
CO02	✓	✓	✓	✓	✓
CO03	✓	✓	✓	✓	✓
CO04	✓	✓	✓	✓	✓
CO05	✓	✓	✓	✓	✓
CO06	✓	✓	✓	✓	✓
CO07	✓	✓	✓	✓	✓
CO08		✓	✓		✓

Bibliography: Highly variable, dependent upon application area selected by student.

CSC 498: Requirements for Project Proposal

First: the onus of picking a topic, developing a proposal and completing the proposal is on the student (or group of students). The supervising faculty member is available for consultation and suggestions, but the student(s) are responsible for "making things happen". Students should not expect specific assigned homework, regular meetings (as in the traditional two or three times a week) and/or tests or quizzes during CSC 498.

Second: CSC 498 and CSC 500 are only offered on a Directed Study basis. In order to register for *any* Directed Study (including CSC 498 and CSC 500), students must fill out a Directed Study Registration form (available from the Registrar's Office). The signatures required include those of the supervising faculty member and the department chairperson. Once the form has been completed, it is the student's responsibility to submit the form to the Registrar's Office. Note that the requirement of the Directed Study form makes it impossible to register for CSC 498 or CSC 500 on-line.

Proposed projects for CSC 498/500 must involve the design and implementation of a moderate-to-large system or software project. Proposals should adhere to the following general guidelines:

- Projects should be primarily applications-oriented and non-trivial in nature; projects must exhibit algorithmic complexity and/or research into area(s) new to the student, and may not be simply "output generators";
- The main focus of the project must draw upon one or more upper-level (above CSC 260) courses, utilizing and possibly extending information (algorithms, structures, methodologies, etc.) acquired in such courses;
- A faculty supervisor must agree to monitor the student's progress and to provide a *limited* amount of technical support. During CSC 498 the supervisor will provide guidance for the student in choosing a topic, designing the proposal, determining the components of the final report and presentation, and creating a proposal presentation; during CSC 500 the supervisor will provide guidance and technical assistance with implementing the project, and creating a suitable presentation of the completed project.
- Once a supervisor has been located and a topic / application area agreed upon, the student must prepare a formal proposal detailing the specific requirements and expectations of the project. **The proposal must include the following components** (explained in more detail below). Proposals lacking any of the following components will not be scheduled for presentation.

- Cover Page
 - Student Objectives
 - Problem Specification
 - Tools List
 - Solution Processes and/or Design
 - Time Schedule
 - Grading Scheme
 - List of Deliverables
 - Presentation (must be in "presentation format", e.g., Microsoft Office PowerPoint, OpenOffice Impress, Prezi, etc.)
-

Cover Page

Center the project title on the page. Place the name(s) of all student participants under the title. Place the name of the faculty supervisor and the presentation date in the bottom right corner.

Student Objectives

State what your personal goals and objectives for the project are, that is, state in general terms what you hope to accomplish by completing your proposal and project, and then state the specific new skills and/or skill enhancements you expect to demonstrate via your project. Examples include "experience with advanced database design concepts", "experience with the complete life cycle of a project, from initial fact-finding and problem specification all the way through to implementation, verification and documentation", "ability to install, configure and use MySQL", etc.

Problem Specification

Describe in clear *non-technical* language what the project will attempt to do. Explain any terms that may be unknown to a reader unfamiliar with the specific subject area of the proposal. Focus on *what* the project will accomplish, that is, on the functionalities that it will support. The problem specification should be one to two pages in length.

Tools List

List any and all tools that may be used in developing a solution to the problem. Tools include (but are not limited to):

- any software or hardware that will be used at any stage of the process, including (but not limited to) program language IDEs, CASE environments, operating system(s), communication protocols, general productivity tools, FPGA kits, etc.;
- algorithms and/or data structures, *if beyond those implemented as part of previous coursework.*

If evaluation and selection of tools is part of project implementation, state so explicitly; include a list of potential candidates and specify the criteria to be used in selecting specific tools.

Solution Design

Describe the proposed high-level (architectural or abstract) design of the solution in a format appropriate to the subject area (e.g., ER diagrams for database-centric projects, UML diagrams (class, activity, interaction and/or use-case) for large-scale software projects). For each major component of the architecture, include a brief paragraph describing the responsibility of the component. *Benchmarks must be defined which will allow progress in the project to be monitored and documented.* The design should not go into implementation details: it should present the upper levels of the system architecture of the proposed solution.

Time Schedule

Establishing a timetable and agreeing on a reasonable rate of progress on the project is the joint responsibility of the student(s) and the faculty supervisor. List the major components/benchmarks from the previous step in the order in which they will be completed. Indicate which (if any) are dependent on earlier steps, and which (if any) can be worked on simultaneously (Gant or PERT charts may be appropriate). Include approximately how much time each component should take (in days or weeks): the total amount of time allocated should be approximately 14 weeks.

Grading Scheme

Possibilities include allocating a percentage of the grade to each of the components / benchmarks of the project, or specifying the set of benchmarks representing progress of the project and awarding a final grade based on how many of the benchmarks have been reached *and documented*. Use the **Time Schedule** list as a reference for the components / benchmarks. The supervisor must approve the final grading scheme. Note that the presentation of the completed project must be allocated 10% of the final grade for CSC 500.

Deliverables

The specific list of deliverables will vary from project to project. Typical deliverable components include, but are not limited to, the following. Note that *all* projects *must* include the components presented in **boldface**.

Not all components will be included in all proposals; additional components may be required at the discretion of the supervisor, based on the nature of the proposed project. **Note that the following list is of components that are to be delivered upon the completion of CSC 500, not CSC 498. In CSC 498, you are just listing what will be included in the completed package.**

- **original proposal (from CSC 498)**
- **amendments to the proposal (approved by the project supervisor)**
- system architecture diagram(s) (UML, DFD context, etc.)
- design diagrams (UML, enhanced ER, etc.)
- fully-commented source code
- compiled source listings
- complete test data sets
- test data results
- sample output (screen shots and/or reports)
- user's manual
- executables and/or projects
- **presentation documents (used to support the presentation of the completed CSC 500 project) a narrative of the progress of the project, in clear, concise English, including any problems**

- encountered and how said problems were addressed
- a summary of what was learned from the project and (based on that experience) discussion of how various aspects of the project might have been approached differently
- a list of what areas of the proposal (if any) were not completed, including reasons why
- presentation of the completed project (PowerPoint format), including screenshots of the functioning project

Not all components will be included in all proposals; additional components may be required at the discretion of the supervisor, based on the nature of the proposed project. Note that the preceding list is of components that are to be delivered upon the completion of CSC 500, not CSC 498. In CSC 498, you are just listing what will be included in the completed package.

Presentation

A slide show-based presentation of the proposal must be created, reviewed by the faculty supervisor, and presented to the Computer Science Department. The presentation must be a *summary* of the proposal, *not* a duplicate of it. In particular, the Project Specification and Solution Design components will always need to be summarized / condensed; the remaining components can sometimes be copied out of the proposal and pasted into the presentation, but will usually also require summarization. There are no set limits on the number of slides - however, note that you will have no more than 15 minutes in which to make your presentation.

A copy of the finished proposal must be submitted to the faculty supervisor at least *one week before the last day of classes for the semester*; copies must be provided for each faculty member of the department on Presentation Day. The student is responsible for distributing the copies to faculty. The finished proposal will be presented by the student(s) to the Computer Science Department on Presentation Day (typically on the Reading Day which immediately precedes the final exam period). Final approval takes the form of a grade of "Pass". Non-approval of a proposal (a grade of "Fail") is to be transmitted to the student(s) by the faculty supervisor, accompanied by a detailed analysis of why the proposal was rejected. Rejected approvals may be re-submitted for consideration during the next semester.

Projects may be disallowed for insufficient technical content, duplication of current or previous projects, or insufficient background on the part of the student. It may be necessary to postpone a project to a future semester due to unavailability of a faculty supervisor. A student may not register for CSC 500 until a grade of Pass has been achieved for CSC 498.

Final reminder: the onus of picking a topic, developing a proposal and completing the project in a timely fashion is on the *student* (or group of students). The supervising faculty member is available for consultation and suggestions, but the student(s) is/are responsible for "making things happen" at all stages of the proposal and project. Students should not expect standard course ingredients such as specific assigned homework, regularly scheduled meetings and/or tests or quizzes.

CSC 500: Requirements for the Implementation of the Proposal

Students must have completed CSC 498 and must fill out a Directed Study Registration form in order to register for CSC 500 (note that this is the same for that was filled out for CSC 498 - it *does* have to be filled out again for CSC 500). Note that since completion of CSC 498 does not take place until the end of the semester (after the formal presentation of the proposal), CSC 500 *cannot* be registered for during advising / pre-registration: **registration for CSC 500 must take place through the Registrar's Office**. Note that the signatures required include those of the supervising faculty member and the department chairperson.

Once work has begun on a project, modifications in the original proposal may be found necessary. Any such modifications must be justified and submitted, in writing, to the faculty supervisor, and subsequently approved by the faculty supervisor before being implemented.

Establishment of the timetable for the project and agreement on a reasonable rate of progress *was* (note the use of past tense!) the joint responsibility of the student(s) and faculty supervisor in CSC 498 when the project proposal was formalized. During CSC 500 it *is* (note the use of present tense!) the responsibility of the *student(s)* to maintain this rate of progress and meet the agreed-upon deadlines. Direct supervision of the project by the faculty supervisor is minimal - there are no regularly-scheduled lectures as in most academic courses. Students may consult with the supervisor for suggestions as to how to approach an unexpected problem or where to go to find technical support; students should *not* expect the faculty supervisor to assist in debugging code or to provide detailed technical assistance. Inadequate progress on the part of any student may, at the discretion of the supervisor, result in a failing grade for that student.

One week **before** the last day of classes for the semester the student(s) must submit to the faculty supervisor all required deliverables for review:

- the original proposal, as approved by the Directed Study Committee;
- any modifications or extensions to the original proposal as approved by the faculty supervisor and the Directed Study committee;
- a narrative of the progress of the project, in clear, concise English, including any problems encountered and how said problems were addressed;
- required deliverables (deliverable components as specified in the final approved proposal from CSC 498)
 - **in particular, the PowerPoint presentation of the completed project (PowerPoint format) must be provided to the supervisor for review well before Presentation Day**
- a summary of what was learned from the project and (based on that experience) discussion of how various aspects of the project might have been approached differently;
- a list of what areas of the proposal (if any) were not completed, and why.

Once the deliverables have been approved by the supervisor, all deliverables must be burned to CD/DVD, with two (2) copies submitted to the supervisor on Presentation Day. In addition, printed copies of the project presentation (including screen shots) must be made available to all department faculty members attending Presentation Day.

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.