# CSc 190 and CSc 191 - SENIOR PROJECT

## Policies and Course Syllabus

Department of Computer Science

School of Engineering and Computer Science

California State University, Sacramento

© R. Buckley - August 24, 2001

## 1.0 INTRODUCTION.

Senior Project provides students with the following opportunities:

- To apply what has been learned in three (or more) years of academic work in computer science to the development of a real-world software product.
- To work as a member of a software development team.
- To provide a learning experience in which software engineering principles are applied to a real-world project.
- To demonstrate individual capabilities in the real world setting.
- To create a portfolio that provides evidence of quality work.

## 1.1 Course Prerequisites.

Students enrolling in Senior Project should have satisfied the following requirements:

- Received a passing grade on the Writing Proficiency Exam (*WPE*)
- Attained senior standing (completing 90 or more units)
- Classified as a Computer Science major or a double major in Computer Science and Mathematics
- Completed a minimum of five Computer Science courses numbered 131 or higher (not including courses numbered 192, 194, 195, 198, or 199).

Note. “Completion” means a grade of C- or better in these courses.

If the student has not attained senior standing or not completed the minimum number of upper division computer science courses, permission may be granted to enroll in Senior Project. However, this requires that the student be two semesters away from graduation, which must be documented with a graduation agreement verifying the graduation date. If a student manages to enroll in Senior Project without meeting the eligibility requirements, that student can be dropped from the class. *If there are any questions about eligibility, see the instructor immediately.*

Students should not take Senior Project too early. While many students reach “senior” status before they are in their final year of studies, they are not adequately prepared to be an effectively contributing member of a senior project team. The senior project requirement is intended to be a culminating experience for the major. Consequently, the typical student should be enrolling in CSc 190 two semesters before graduation.

(Revised 8/26/01-BB)
2.0 COURSE DESCRIPTION.

CSc 190/191 is a two-semester course sequence, which is structured as a laboratory experience, providing computer science majors the opportunity to apply the knowledge acquired during three (or more) years of college study to develop a real world software product. Each student is required to work as a member of a project team. The team must obtain a sponsor and a project, which must be approved by the senior project instructor. The sponsor serves as the customer for whom the software product is being developed. Sponsors may be representatives from either the private or public sector, for example, a local business, a government agency, a non-profit entity, or University.

The product is developed using a formalized, software engineering development process. In this process each team completes the necessary software engineering work and publishes this technical work in a set of documents. Completion of each document serves as a major milestone representing the culmination of the technical work completed in each phase in the life cycle of the project’s development. During the first semester (CSc 190) teams are formed, an acceptable project is identified, and, upon approval, to work begins. The work requires an initial proposal to be prepared, a project management plan completed, the analysis and specification of the software requirements collected and documented. All work is reviewed by the team’s faculty adviser prior to the approval. During the second semester (CSc 191) the teams design the software, construct the product according to these design specifications, prepare a system test plan, and – using the plan - test and deliver the product along with appropriate user documentation.

Completion of senior project requires delivery and acceptance of the product by the project sponsor and the delivery of the complete set of project documents to the senior project instructor and to the project sponsor.

2.1 Some References.


2.2 **Project Emphasis and Expectations.** The primary purpose and intent of the senior project requirement is to use a relatively small project to simulate the software engineering practices needed in the development of a much larger software system involving a much larger project team. Students should understand that, no matter how small the project, teams are required to follow a standard software development process, one that establishes requirements before the product is designed, produces a design specification that accurately represents the product as implemented, plans and specifies the system testing needed prior to testing, etc. Teams will spend more time on the “front end” activities of the project (requirements and design) than would normally be expected with a small project.

Most importantly, the project team as well as the individual team members must demonstrate that the project was effectively managed from beginning to end on a week-by-week basis. In addition, the team must demonstrate that all members were engaged in this weekly work and that each contributed their fare share to the project. Upon completion of the project and the course, the teams as well as their members should be able to demonstrate an understanding of the discipline and practice necessary to successfully develop and deliver a software product.

2.3 **Project Life Cycle.** Table 1 provides a listing of the five required phases of the project life cycle and each phase’s associated activities. The distribution of work effort is estimated using actual times from previous Senior Projects.
Table 1: Required Software Development Process

<table>
<thead>
<tr>
<th>DISTRIBUTION OF WORK</th>
<th>Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phase and Activities</strong></td>
<td><strong>Phases</strong></td>
</tr>
<tr>
<td>System Requirements Analysis</td>
<td>5%</td>
</tr>
<tr>
<td>Identify project</td>
<td></td>
</tr>
<tr>
<td>Prepare project proposal</td>
<td></td>
</tr>
<tr>
<td>Project Management Plan</td>
<td>5%</td>
</tr>
<tr>
<td>Prepare WBS</td>
<td></td>
</tr>
<tr>
<td>Prepare Schedule</td>
<td></td>
</tr>
<tr>
<td>Prepare SPMP</td>
<td></td>
</tr>
<tr>
<td>Software Requirements Generation</td>
<td>14%</td>
</tr>
<tr>
<td>Complete requirements analysis</td>
<td></td>
</tr>
<tr>
<td>Prepare technical specification</td>
<td></td>
</tr>
<tr>
<td>Prepare SRS</td>
<td></td>
</tr>
<tr>
<td>Software Design</td>
<td>11%</td>
</tr>
<tr>
<td>Prepare architectural design</td>
<td></td>
</tr>
<tr>
<td>Prepare detailed design</td>
<td></td>
</tr>
<tr>
<td>Prepare SDD</td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td>17%</td>
</tr>
<tr>
<td>Write code</td>
<td></td>
</tr>
<tr>
<td>Perform unit testing</td>
<td></td>
</tr>
<tr>
<td>Perform integration testing</td>
<td></td>
</tr>
<tr>
<td>System Testing</td>
<td>6%</td>
</tr>
<tr>
<td>Prepare system tests</td>
<td></td>
</tr>
<tr>
<td>Perform system testing</td>
<td></td>
</tr>
<tr>
<td>Prepare STD and STR</td>
<td></td>
</tr>
<tr>
<td>User Documentation</td>
<td>3%</td>
</tr>
<tr>
<td>Design and write some form of user documentation</td>
<td></td>
</tr>
<tr>
<td>Learning</td>
<td>22%</td>
</tr>
<tr>
<td>(all phases)</td>
<td></td>
</tr>
<tr>
<td>Project Management</td>
<td>18%</td>
</tr>
<tr>
<td>(entire project)</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>100%</td>
</tr>
</tbody>
</table>

2.4 **Student Responsibilities.** The two-semester sequence takes the student step-by-step through the development and delivery of a software product. The lecture portion of the course provides instructional guidelines on what is to be done. However, the lectures will not provide everything needed to successfully complete the entire software engineering project. The student’s previous three (or more) years of computer science education should have provided much of what is needed to be successful in Senior Project. In fact, Senior Project provides an opportunity for the student to apply the knowledge accumulated during their academic “career”
as a computer science major. However, there will be “gaps” which the student will be required
to fill, and this may require a considerable amount of ad-hoc, on-the-job learning.

Students are expected to have (or develop) a working knowledge of the following:

- Preparing and writing technical reports
- Preparing and presenting oral technical reports
- Planning and scheduling an activity
- Analyzing a system and specifying its requirements
- Depicting requirements using some representational methodology (e.g. UML)
- Designing a system from requirements specifications
- Depicting a system design using a representational methodology (e.g. UML)
- Building a system from a design specification
- Using an acceptable programming language(s) and tools to implement the software design
  specifications
- Developing a test plan and designing appropriate test cases sufficient for system testing
- Testing the system using a system test specification
- Applying quality assurance principles and practices
- Performing technical reviews
- … or whatever else is needed.

Deficiencies in any of the above areas must be corrected through self-study.

3.0 THE PROJECT.

The Senior Project class is intended to help students make the transition from academic life to
professional life. The project will require students to assume responsibilities broader in scope
than one would normally have the opportunity to assume even in their early years of
employment.

3.1 Project Responsibilities. In a typical classroom setting, all students do the same
assignment or project. The instructor provides the proper guidance or the means to accomplish
the assignment. The approach to be taken is either specified in the curriculum or text, or
previous examples of similar assignments are available. The instructor is not a “customer” who
depends on the student’s successful completion of the assignment. In fact, the consequences of
not completing the assignment are reflected only in the student’s grade.

In the senior project course the assignment is a real world project. If such a project has been
done previously, there is seldom any need to do it again. Furthermore, if no one is depending
upon the project being completed, there is no need to do it at all. Hence, the project for this
course must be something that has not been done, will have some risks, and has no guarantees
that it will be completed on the prescribed “due date”. Furthermore, in some cases there is no
guarantee that the project is doable or that the resources needed to complete the project will be
available. However, someone should be depending upon the completion of the project. This is
the most significant difference.

Student teams must assume the responsibility to keep their customer informed as to how the
work is proceeding and what the expectations are that the project will be completed as planned. If expectations change, delivery of a satisfactory product within the time allotted may not be possible. In such a case, the project team must assume the responsibility to negotiate with their customer whatever changes are necessary to ensure the delivery of the product.

Just as in the private and public sector, each student will be expected to perform professionally. As a member of a team, they will be responsible for establishing schedules and setting goals. The teams will also be expected to meet these goals and achieve a high standard of product quality.

3.2 Project Requirements. Each project will require in excess of 200 hours of work per student - although most projects require considerably more time. The project is expected to extend over a full two semesters. To gain credit for CSc 190, the project team must satisfy all requirements for each of the deliverables listed in Table 2 as well as fulfilling the expectations specified in section 2.2 above. Students not achieving at least a grade of C- in the first semester will NOT be allowed to continue individually or as a team and must reenroll in CSc 190. To gain credit for CSc 191, the project team must satisfy all the requirements for each of the deliverables listed in Table 2 as well as fulfilling the expectations specified in section 2.2 above.

Once the project schedule has been established, the project team’s faculty adviser must approve any changes in this schedule and/or sequencing of the deliverables. However, students should understand that all requirements for CSc 190 and CSc 191 must be completed by the end of the respective semester.

Table 2a: Sample Project Schedule for CSc 190

<table>
<thead>
<tr>
<th>CSc 190 Activity</th>
<th>Deliverable</th>
<th>Week Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify Project</td>
<td>Project Abstract (formally initiates the project)</td>
<td>?</td>
</tr>
<tr>
<td>Prepare Proposal</td>
<td>Software System Proposal (SSP)</td>
<td>?</td>
</tr>
<tr>
<td>Plan Project</td>
<td>Software Project Management Plan (SPMP)</td>
<td>?</td>
</tr>
<tr>
<td>Determine Software Requirements</td>
<td>Requirements Review and Sign-off</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>Software Requirements Specification (SRS)</td>
<td></td>
</tr>
<tr>
<td>Prepare for Customer Presentation</td>
<td>Project Status and SRS Verification Review</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Presentation</td>
<td></td>
</tr>
</tbody>
</table>
Table 2b: Sample Project Schedule for CSc 191

<table>
<thead>
<tr>
<th>CSc 191 Activity</th>
<th>Deliverable</th>
<th>Week Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete System Design</td>
<td>Software Design Description (SDD)</td>
<td>?</td>
</tr>
<tr>
<td>Prepare System Tests and Test Cases</td>
<td>Software Test Description (STD)</td>
<td>?</td>
</tr>
<tr>
<td>Implement System, Conduct Unit and</td>
<td>Completed Coding (CODE)</td>
<td>?</td>
</tr>
<tr>
<td>Integration Testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conduct System Tests</td>
<td>Software Test Report (STR)</td>
<td>?</td>
</tr>
<tr>
<td>Create User’s Manual</td>
<td>User’s Manual (UM)</td>
<td>?</td>
</tr>
<tr>
<td>Prepare Maintenance Manual</td>
<td>Software Maintenance Manual (SMM)</td>
<td>?</td>
</tr>
<tr>
<td>Prepare for Customer Presentation</td>
<td>Delivery: installation, demonstration, and</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>SMM</td>
<td></td>
</tr>
</tbody>
</table>

3.2.1 Preliminary Project Schedule. Each team is required to prepare and submit to the instructor a schedule of activities and deliverables (Tables 2a and 2b can be used as a guide in this task). This schedule should be expanded, showing weekly work assignments and then documented in the team’s Project Management Plan. Any revisions in this schedule should be clearly specified, submitted to the instructor and attached as updates to the Management Plan.

3.2.2 Project Documents. A typical Senior Project team will prepare the following documents (Table 2c) in approximately the order given. Baseline documents are considered complete after all indicated parties have approved the document and signed the delivery page. The “yes” entries indicate the minimum number of signatures required.

Table 2c: Sign-off Requirements

<table>
<thead>
<tr>
<th>Baseline Document</th>
<th>Team Members</th>
<th>Team’s Faculty Project Adviser</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Project Proposal (SSP)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Software Project Management Plan (SPMP)</td>
<td>yes</td>
<td>yes</td>
<td>optional</td>
</tr>
<tr>
<td>Software Requirement Specification (SRS)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Software Design Documents (SDD)</td>
<td>yes</td>
<td>yes</td>
<td>optional</td>
</tr>
<tr>
<td>Software Test Description (STD)</td>
<td>yes</td>
<td>yes</td>
<td>optional</td>
</tr>
<tr>
<td>Source Code Listing Document (CODE)</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Software Test Report (STR)</td>
<td>yes</td>
<td>yes</td>
<td>optional</td>
</tr>
<tr>
<td>User’s Manual (UM)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Software Maintenance Manual (SMM)</td>
<td>yes</td>
<td>yes</td>
<td>yes*</td>
</tr>
</tbody>
</table>

- represents acceptance of the delivered software product.

NOTE. Any subsequent changes in the technical specifications require that updates be prepared
and changes approved by the indicated parties prior to formally updating any of these baseline documents.

3.2.3 Additional Deliverables. Individual lab instructors may require teams to complete additional deliverables during various phases of the project. The intent, in all cases, is to provide for a more structured sequencing of activities as teams plan and work toward completion of the major milestones. Examples might be the completion of the work breakdown structure chart prior to completion of the project management plan, or the completion of UML diagrams prior to the completion of the software requirements specification document.

3.2.4 Oral Presentation. Students can expect during their professional career to be called upon to attend many presentations as both a participant and presenter. As a presenter they will be expected to effectively communicate and, in many cases, to seek feedback from management, coworkers, subordinates, customers and other groups. The content of these presentations will vary, at times being focused on technical content while at other times being focused on ideas, proposals and recommendations. Those that are able to effectively organize information and have developed good presentation skills will have an advantage over those that lack such skills. Such skills add significant value to the organization and consequently those with the skills are more apt to receive career advancement opportunities, including more challenging job assignments, increased authority, responsibility, and remuneration.

3.2.4.1 Objectives. The presentations required of Senior Project teams satisfy the following objectives.

- To design and deliver a formal presentation on technical aspects of the project
- To design and make a formal presentation on the project’s status
- To design and fabricate visual aids to support the presentation
- To receive and formally reflect on feedback from members of other project teams on the effectiveness of the presentation

Each semester project teams will be required to give two presentations, one during class and one in a meeting with the customer. In addition, the lab instructor or the customer may require additional presentations.

3.2.4.2 Content of Customer Presentations. As indicated above, teams are required to give, at a minimum, two oral presentations to the customer. The first presentation is called a software requirements review and will contain information on both the technical aspects as well as the managerial status of the project. In addition, a prototype of the proposed product may be presented to facilitate feedback on the requirements as specified. This presentation is given at the end of the first semester (CSc 190). The second presentation is given at the end of the second semester (CSc 191) and marks the completion of the project. This presentation typically includes a summary of the team’s work and a discussion of the product test results along with the installation and demonstration of the product.

3.2.4.3 Evaluation. Those in attendance will be given a rating sheet to provide feedback to the presenting team as to style, presentation context, visual aids, manner of speaking, etc. In addition, instructor and/or project faculty adviser will provide feedback.
Note. In the case of the classroom presentations, those not presenting will provide feedback on the team’s presentation.

3.3 Selecting a Project. It is difficult to select a project that will be of sufficient scope. As indicated above, most Senior Projects have proven to require more time than initially estimated. However, those projects that do not measure up to the original estimate of scope of work may need to be reduced (or, if necessary, increased).

Warning. A penalty will be assigned to those teams that require a reduction in the scope of the project merely because they have not committed the necessary time and effort and, consequently, have not effectively managed their work.

3.3.1 Identifying an Acceptable Project. The following are some guidelines for identifying an acceptable project:

- The project cannot be simply a game, or a “home brew” project.
- The project must be sponsored by an outside, independent customer.
- The project must be “big” enough for a 5 to 6 person team.
- The project must require teams to go through all phases of a software development life-cycle.
- A typical project requires the implementation of a client-server architecture containing a data base requirement.
- The project must be big enough to involve each team member in all aspects of the development.
- The project must be big enough so that there is some risk of failure.

3.3.2 Identifying an Acceptable Customer. The following are the restrictions on selecting a project customer/sponsor.

- The customer/sponsor must not be related to student.
- None of the project team may work directly for the customer/sponsor.
- The customer/sponsor must represent a legitimate business, governmental entity, a charitable organization, or an educational institution.

3.3.3 Contracted Responsibilities. The Software Project Proposal and the Software Requirements Specification are contracts signed by the student, the customer and the instructor. Since the project, by definition, is one that has never been done before, it is likely that unforeseen problems will arise. Therefore, the student, customer, and instructor must all understand that the original contract is subject to re-negotiation. If the original work has been planned well, the schedule monitored on a weekly basis, and the status kept and reported, then problems can be identified and the project schedule can be easily modified.

Warning. Again, the option of modifying or reducing requirements may not be extended to students who have not kept their plans and status current.

3.4 Project Manager. Each student will work on a project team headed by one student
identified as project manager or project leader. The project manager is responsible for ensuring that work assignments are allocated to all team members and that problems (technical and managerial) are dealt with in a timely manner – involving the team’s project faculty adviser when necessary. The project manager is also responsible for keeping track of and reporting the time allocated and spent by each team member on each task assigned.

3.5 Manager of Projects. As manager of projects, the team’s faculty advisor (the lab instructor) wears two hats. Like a “division” manager for the lab’s software projects, he or she is responsible for the successful delivery of the products being developed by each of the teams under his or her supervision. In the role as manager, he or she will insist that students and teams comply with project procedures and policies. In the role as instructor, he or she will provide the guidance and feedback necessary to facilitate the student’s learning of real-world software engineering principles.

3.6 Team Formation. In some cases teams are formed before the first meeting of CSc 190. These teams will be formally constituted at the first meeting. For those students who have not formed into teams, they will be assigned to a team at the first class meeting.

Although many of the entering students will not be able to select their team members, doing so can be very critical to the success or failure of the project. A team can be more effective if there is a diversification of talents among team members. However, in forming a team or deciding to join a team, consideration should be given to answers for the following questions. Are the technical skills of the team members equal and/or balanced? Are their work habits compatible? Can each member contribute his or her fair-share of the work? Will the team continue to perform effectively under stress? Is there a potential for personality conflicts over the course of one to two semesters? Are team members all committed to producing a quality product? It is important for all team members to commit to the whatever the necessary work will be over the next 30 weeks. The project’s success requires that such a commitment be made by each team member.

3.7 Project Schedule and Status. The project team will include a tentative schedule in their Software Project Proposal and the actual baseline schedule in the Software Project Management Plan. In addition - as indicated above - the team is responsible for collecting and reporting data on person-hours worked during each phase of the project – week by week. Recording and reporting will begin immediately after the Project Abstract has been approved by the lab instructor.

3.8 Legal Disclaimer. Neither California State University, Sacramento, the instructors, nor any of their agents bear any responsibilities for any damages arising from any project undertaken by students in Senior Project.

3.9 Ownership of the Product. As students registered in the senior project courses, team members neither employed by California State University, Sacramento or the project’s sponsor. Consequently, the team, collectively, “owns” the final product and all project related materials. However, if the sponsor or any other interested party requires joint ownership and/or clear legal title to the product, a separate agreement should be prepared and signed. If this is the case, the Department recommends that the parties seek the advice of a qualified and competent lawyer.
In either case, the Software System Proposal should spell out in exact detail all ownership requirements for the project. In any event, the Computer Science Department reserves the right to use both the documentation and the product as examples of student work.

4.0 DOCUMENTATION.

All technical work – from requirements analysis through to final delivery – will be documented. Guidelines will be provided for the preparation of each document.

Each document should be self-contained and written for anyone with a general computer science background. The team should not assume the reader has a detailed understanding of the project.

4.1 Software Maintenance Manual (SMM). At the end of CSc 191, the team will deliver to the customer/sponsor and the Senior Project Seminar instructor a CD that contains copies of all the project documents, the software produced along with instructions for installation of the software. This compendium of project documents (1) represents the final product of the class, and (2) serves as an example for future classes. The SMM is a separate document, which contains an introduction and a brief summary of each document. Each team will also deliver to Senior Project Seminar instructor the hard copy originals of the signed approval pages for each the documents. Failure to deliver the SMM can result in failure to pass CSc 191.

4.2 Document Review and Feedback. The project team is responsible for the quality of the documentation produced. The process used to insure adequate review and revision of the team’s work products will be documented in their Project Management Plan.

The team’s project faculty adviser must approve all the documents produced by the team – and will make every effort to complete the review of submitted documents in a timely manner. In general, the team should expect a one-week turnaround for all materials submitted. However, as deadlines approach, teams should plan for this review time and not expect turnaround time to necessarily fit their scheduled needs.

5.0 COURSE CONDUCT.

The course consists of a weekly lecture/seminar and a weekly “lab” period.

5.1 Lecture/seminar. The lecture/seminar meets in a 75 minute session once a per week and attendance by all team members is mandatory. Over the course of the semester the discussions will include issues related to the software development process as well as topics specific to the work required to successfully complete the project. In addition, each team will give a technical presentation during the seminar – in both CSc 190 and CSc 191.

5.2 Lab Period. In addition to the seminar, students must register for the 150-minute “lab” period. While each team will be assigned to a project faculty adviser (the “lab” instructor) and will meet together each week, the weekly meeting time will not necessarily be scheduled during the 150 minute designated time. Each team will consult with their faculty adviser and establish a regular, weekly meeting time. Attendance at these meeting by all team members is mandatory.
The following list provides an example what might be expected during this period.

- Discuss the work completed during the previous week
- Discuss problems
- Turn in the staff-hour accounting sheet and schedule updates
- Review work assignments for the coming week
- Review project faculty adviser’s comments on submitted document drafts
- Negotiate changes in project schedule

5.3 Drops, Incompletes, and Cheating. All students enrolled in the senior project sequence are subject to the University and the Department of Computer Science policies regarding drops, incompletes, and cheating. Copies of these policies may be obtained in the Department office.

Cheating in CSc 190/191 is defined as (1) submitting someone else’s specific work (i.e., without crediting the source), (2) forging a signature, and (3) intentionally submitting erroneous reports. However, copying from documents that are essential to the support and development of the project is permissible. If this information is general, credit need not be given, however if the information is specific credit must be given. If students have questions about what is permissible, consult the lab instructor.

5.4 Grades. Both CSc 190 and CSc 191 includes a separate grade for the seminar portion of the course and separate grade for the lab portion of the course (see Table 3 below).

The seminar grade, assigned by seminar instructor, is a reflection of the student’s attendance, participation in classroom activities, completion of assignments, and the team’s submission of weekly status information along with whatever other reports are required.

The lab grade, assigned by the team’s project faculty adviser, is a reflection of the student’s performance as a member of the project team and his or her contribution to the development effort.

The student’s grade for CSc 190 or CSc 191 is a combination of the seminar and lab grade weighted as indicated in Table 3.

<table>
<thead>
<tr>
<th></th>
<th>CSc 190</th>
<th>CSc 191</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seminar</td>
<td>15%</td>
<td>15%</td>
</tr>
<tr>
<td>( 1.5 grade points )</td>
<td>85%</td>
<td>85%</td>
</tr>
<tr>
<td>Lab</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>TOTAL</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

The following general criteria will be used to assess team and individual performance (see CSUS

**A** is given for exemplary achievement of the course objectives. In addition to being clearly and significantly above the requirements, work exhibited is of an independent, creative, contributory nature.

**B** is given for superior achievement of the course objectives. The performance is clearly and significantly above the satisfactory fulfillment of course requirements.

**C** is given for satisfactory achievement of the course objectives. The student is now prepared for advanced work or study.

**D** is given for unsatisfactory achievement of course objectives, yet achievement of a sufficient proportion of objectives so that it is not typically necessary to repeat the course. Note. As a major in Computer Science, all courses taken to fulfill your major require C- or better and repeating such a course would be required.

**F** is given for unsatisfactory achievement of course objectives to an extent that the student must repeat the course to receive any credit.

Students that receive a D or F in either CSc 190 or CSc 191 are required to repeat senior project, beginning again with CSc 190.

Note. In the “real world”, a project and those working on the project are subject to review by management. Such reviews have an affect upon the career, the salary and even the continued employment of those involved in the project. Since students are not (and cannot be) paid for Senior Project work, each student’s “salary” is received in the form of a grade. “Continued employment” should be interpreted as a passing grade in both CSc 190 and CSc 191.

### 6.0 PROFESSIONAL SOCIETIES.

Finally, each technical discipline has a number of professional societies and publications that exist to provide a forum for presenting and disseminating professional views and advancements in the state-of-their-art. As a computer scientist or software engineer, membership in such professional organizations is common. The Association for Computing Machinery (ACM) and The Institute for Electrical and Electronic Engineers Computer Society (IEEE-CS) are the two most highly respected computer organizations are. The School of Engineering and Computer Science is fortunate enough to have a joint ACM/IEEE Computer Society student chapter. Students interested in joining should contact the chapter’s faculty advisor or visit the student chapter homepage ([http://www.ecs.csus.edu/students/acm](http://www.ecs.csus.edu/students/acm)).