CSIS 4485: Software and Security Engineering
Sections 001 and 002
Fall 2016: Syllabus

Instructor:	Dr. Vincent Cicirello	**Office:**	G-116
E-mail:	cicirelv@stockton.edu	**Phone (office):**	609-626-3526

Office Hours: Tuesdays & Thursdays: 1:15pm-2:15pm (available other times by appointment)
Or, feel free to drop-in any time I'm in my office (if I'm there, I'd be happy to talk to you).

Course Time and Location: Section 001: Tuesday and Thursday, 2:30pm-4:20pm
Section 002: Tuesday and Thursday, 8:30am-10:20am
Location: Tuesday: D003 / Thursday: D017 (computer lab)

Course Description: An introduction to software engineering and security engineering in software systems. The course addresses universal software engineering techniques as well as techniques for development of critical software systems where the cost of system failure is potentially high. Software engineering topics covered will include software process models, agile software development methods, requirements engineering, system modeling, architectural design, implementation, software testing, and evolution. Topics related to software dependability and security engineering are also covered.

Prerequisites: CSIS 3103: Data Structures or (CSIS 3222 and CSIS 3470).

Required Textbooks/Readings: Ian Sommerville, Software Engineering, 10th Edition, 2016. ISBN: 0-13-703515-2.

Grading:		
	Exam 1 / Exam 2	15% / 15%
	Course Project	45%
	Lab Assignments	10%
	Project Reflection Essay	10%
	Participation	5%

Grading Scale:

A: Overall average at least 90.00 **AND** Exam average at least 80.00.	A-: Overall average at least 89.00 **AND** Exam average at least 80.00.	B+: Overall average at least 88.00 **AND** Exam average at least 80.00.
B: Overall average at least 80.00 **AND** Exam average at least 70.00.	B-: Overall average at least 79.00 **AND** Exam average at least 70.00.	C+: Overall average at least 78.00 **AND** Exam average at least 70.00.
C: Overall average at least 70.00 **AND** Exam average at least 60.00.	C-: Overall average at least 69.00 **AND** Exam average at least 60.00.	D+: Overall average at least 68.00 **AND** Exam average at least 60.00.
D: Overall average at least 60.00 **AND** Exam average at least 50.00.	D-: Overall average at least 59.00 **AND** Exam average at least 50.00.	F: Overall average less than 59.00 **OR** Exam average less than 50.00.

Academic Honesty: Please familiarize yourself with Stockton's policy on academic honesty. Any first violation will be penalized by a 0 on the relevant assignment/exam/etc, plus a 10 point penalty on your overall course grade. Subsequent violations will result in a course grade of F. Examples of violations include, but are not limited to: (a) any form of cheating on an exam or assignment, (b) passing off the work of another as your own, (c) assisting someone in violating the academic honesty policy, (d) asking someone to assist you in cheating or other academic honesty violations (even if they refuse to help you cheat), etc.

Incomplete Policy: In general, no grades of incomplete will be given. The only exception to this rule is an institutionally documented medical emergency that necessitates your complete absence from Stockton for at least two continuous semester weeks. Additionally, you must be caught up on all work up to the point where your medical emergency began and currently in the "C" range or better overall at the point where the emergency began.

Important Dates:

- No Class: Precepting Day (October 25) / Thanksgiving (November 24).
- Demo of Project to Class: Thursday, December 8 during class.
- Deadline for Final Versions of all Project Deliverables: Friday, December 9, 11:59pm, electronically.
- Deadline for Project Reflection Essay (individual assignment): Monday, December 12, 11:59pm, electronically.
- Exam 1: On or about October 18 (will give you a more firm date in a couple weeks).
- Exam 2: Tuesday, December 6 (yes, really not on the last day, deliberately).

IDEA Objectives:

- IDEA Objective 1: Gaining knowledge of the terminology, methods, and trends of software engineering and of security engineering for software systems.
- IDEA Objective 2: Learning the fundamental principles and theories underlying the engineering of dependable software systems.
- IDEA Objective 3: Learning to apply software engineering principles to the development of software systems.
- IDEA Objective 4: Developing the skills needed by professional software engineers and other software development professionals.
- IDEA Objective 5: Acquiring skills in working with others as a member of a team.

CSIS Learning Outcomes: This course supports several of the CSIS Program's Learning Outcomes and corresponding Performance Indicators, as follows:

- Outcome CSIS.a: An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline.
  - CSIS.a.3: Students will apply discrete mathematics concepts and algorithms.
- Outcome CSIS.b: An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
  - CSIS.b.1: Students will analyze the computing requirements for a given problem description.
  - CSIS.b.2: Students will identify the resources, key components, and algorithms required to solve a given problem.
- Outcome CSIS.c: An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
  - CSIS.c.1: Students will design a computer-based solution for a given problem description.
  - CSIS.c.2: Students will implement a computer-based system, process, component, or program from a given specification.
  - CSIS.c.3: Students will evaluate a computer-based system, process, component, or program to assess its conformance to a given specification.
- Outcome CSIS.d: An ability to function effectively on teams to accomplish a common goal.
  - CSIS.d.1: Students will research and gather information.
  - CSIS.d.2: Students will fulfill duties of team roles.
  - CSIS.d.3: Students will share in the work of the team.
  - CSIS.d.4: Students will listen and communicate with other teammates.
- Outcome CSIS.e: An understanding of professional, ethical, legal, security and social issues and responsibilities.
  - CSIS.e.1: Students will recognize and evaluate ethical issues involved in a professional setting.
  - CSIS.e.2: Students will recognize and describe current issues in security.
  - CSIS.e.3: Students will demonstrate understanding of intellectual property issues.
  - CSIS.e.4: Students will recognize the need for proper etiquette and proactive social behavior in professional settings.
- Outcome CSIS.f: An ability to communicate effectively with a range of audiences.
  - CSIS.f.1: Students will write technical documentation of a computer-based system, process, component, or program.
  - CSIS.f.2: Students will make oral presentations for an appropriate target audience.
  - CSIS.f.3: Students will prepare materials for a non-technical audience.
- Outcome CSIS.i: An ability to use current techniques, skills, and tools necessary for computing practice.
  - CSIS.i.1: Students will use a professional integrated development environment (IDE) for implementing programming projects.
  - CSIS.i.2: Students will research online resources to learn and utilize new techniques, skills, and tools.
- Outcome CS.j: An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
  - CS.j.1: Students will apply computer science theory in the modeling and design of computer-based systems.
  - CS.j.2: Students will evaluate the effects of alternative data representations and algorithms on the performance of computer based systems.
- Outcome CS.k: An ability to apply design and development principles in the construction of software systems of varying complexity.
  - CS.k.2: Students will construct a more sophisticated software system using advanced design and development principles.

Software Engineering Topics Covered: The course will cover the following topics, not necessarily in this order:

- Introduction to Software Engineering (Sommerville, Chapter 1)
- Software Processes (Sommerville, Chapter 2)
- Agile Software Development (Sommerville, Chapter 3)
- Requirements Engineering (Sommerville, Chapter 4)
- System Modeling (Sommerville, Chapter 5)
- Architectural Design (Sommerville, Chapter 6)
- Design and Implementation (Sommerville, Chapter 7)
- Software Testing (Sommerville, Chapter 8)
- Dependability Engineering and Security Engineering (subset of Sommerville, Chapters 10, 11, 13 depending on time)

Due Dates: All due-dates are strict. Most project deliverables will have versions due throughout the semester with the final version's grade taking significant precedence over that of earlier versions. In fact, provided the versions at various checkpoints meet certain minimum requirements (will be provided as an additional handout), the only grade that matters will be the grade on the final versions. It is to your great advantage that you meet each of the checkpoint versions to maximize feedback from me earlier in the semester.

Exams: The exams are not cumulative. They are closed book. You are allowed 2 pages of notes on standard letter sized paper (8.5" by 11") for each exam. **Important: The exams can include questions that test your knowledge of any required textbook readings, even if not explicitly covered in class.** Be sure to read the entirety of covered textbook chapters.

Make-Up Exams: Make-up exams in general will not be given (i.e., missed exam = 0), with the following exceptions:

1. Documented medical excuse: provide a doctor's note the first class you return after the missed exam.
2. Other institutional excuses: Situations may arise related to Stockton that prevents you from being able to attend an exam. In most such cases, you should be aware of the conflict beforehand. Thus, I must be notified one week prior to the missed exam. Send me e-mail via Blackboard with the details of the planned absence, and provide me with proper documentation (e.g., memo from sports coach, from other faculty sponsoring a field trip, etc).

Participation: The participation part of your overall course grade is not free points. You are expected to fully participate in all course activities. You will not be directly penalized for an occasional absence (e.g., a small number of absences will not alone reduce your participation grade). **Things that can positively impact your participation grade include:** asking/answering questions during class and other general class participation, assisting others stuck on any part of lab activities, or any other similar things that positively benefit the class. **Things that can negatively impact your participation grade:** frequently skipping class, frequently arriving late or leaving early, sleeping during class, any disruptive behavior (e.g., ringing phones, answering or making phone calls during class, arriving late in a noisy or otherwise disruptive way, leaving early in a noisy or otherwise disruptive way, etc). **Skipping the team project presentations will result in a participation grade of 0** (unless you have a documented medical excuse).

Lab Assignments: Days when we meet in the lab will be of one of 3 types: (a) new material coverage, (b) designated time to meet with your team for project related activities, and (c) lab assignments. The lab assignment portion of your overall grade refers only to this 3rd category. Most of these lab assignments will be individual assignments. There may be some where you can work with another student. And there will be at least one that involves cooperation of your project team members, though it will be graded individually. If you are unable to complete a lab assignment during the normal class time, you must complete it within one week.

Project Reflection Essay: Although your project is a group project, each of you will individually write an essay reflecting upon that project experience. Specifically, this essay will touch upon how your academic learning experience (not limited to this course) prepared you for your course project as well as how your project further enhanced your academic learning. This reflection is an individual assignment. More details on the expectations of this essay will be provided.

Course Project: The course project is a group project with groups comprised of either 4 or 5 members. Your project will take you through the entire software design and development process, beginning with requirements engineering, through system modeling, architecture design, and implementation. The project has several deliverables which will include:

- **Project Proposal:** Once teams are formed, you will have a short period of time to develop a project proposal, which must be approved in advance. During this approval process, I will either: (a) approve the proposal as is, (b) approve with the addition of my own constraints to your proposal (e.g., additional required features of your system, etc), or (c) reject (with reason) and require a new proposal. Most likely reason for rejection is too small of a project. Remember, this is the primary workload for the course, and is a project for 4 or 5 juniors and seniors (mostly seniors).
- **Software Requirements Document:** More detail on this will follow during the first several weeks of the course. The general structure of this document is as outlined in Figure 4.17 (page 114) of the textbook and will include specification of system requirements, system models, system architecture, etc. Various versions will be due throughout the semester. This document will evolve along with the system you are developing itself.
- **System Prototype:** A prototype of the system (e.g., perhaps the user interface, with minimal operational ability) partway through the semester. The system prototype must be consistent with the current state of your requirements document (e.g., consistent with the system models and architecture).
- **Final Version of Software System:** A final working version of the system, which meets the functional requirements your team defined. This version should be consistent with the final version of your requirements document (e.g., consistent with the system models and architecture). All source code for the system must be submitted, including unit testing code and testing data.
- **Activity Log:** Your team must keep records of your activities. When you have meetings, have a team member keep minutes of the meetings (who was there, what was the agenda, what decisions you made, what tasks were assigned to which team members). Report to each other your progress on tasks that you are responsible for, any open issues, etc.
- **Teamwork Peer Evaluations:** I will provide you with a form on which you will be evaluating the members of your team, specifically on teamwork and contributions to the group. I will use these in combination with my own observations in assigning teamwork grades. Note: The teamwork portion of your project grade will likely vary by team member. For all other parts of your project grade other than teamwork, all team members will receive the same grade, with exceptions for grade adjustments per your team's contract (see below).

Team Contract: Each of the project deliverables will receive 1 grade from me for the team. However, individuals on the team may or may not receive that grade. Depending upon the details of a team contract, which you will enter into with each other prior to the start of the project, your individual grade on project deliverables may be adjusted either upward or downward from the grade given that deliverable. Each team will develop, sign, and submit a contract that will establish the structure and expectations of members of your team. This contract provides an opportunity for your team to specify methods of communication, meeting schedules, goals, and consequences of actions (or inactions) of group members. I will provide you with a contract template. Upon group assignment, your team will need to complete, sign, and date the contract and return to me by the specified date. At the end of the semester, your team will then provide me with a follow-up form, signed by all team members certifying any adjustments needed to the distribution of project credit.

Grading of Course Project: The following will be used to combine your grades on the project deliverables into your overall project grade. Note that grades on all project elements listed below (except for "Teamwork") are after applying any adjustments required by your Team's team contract.

Project Proposal	10%
Software Requirements Document	25%
Prototype	10%
Final version of software system	35%
Activity Log	10%
Teamwork	10%

Note: *If you do not show up for the Team project demonstrations, your individual Teamwork grade will be 0* (unless you have a documented medical excuse).

Project Content, Scope, and other Constraints: Your project team will specify, design, implement, and test a significant software system. To provide a common focus for the teams, your system must be an educational tool for a topic of Computer Science of your choosing. Your system must have a Graphical User Interface (GUI), and cannot simply be a text based console application. Your system must be interactive (e.g., feedback to the user on progress, or perhaps adapting to the user's progress), and cannot simply convey static information to the user. Your system should retain data on user's progress across runs, and ideally use that data in some way (e.g., perhaps basing choice of presented problem on user's past performance). It is up to you if you want to handle multi-users. You may develop for Windows or Linux, or cross platform or even a web app if you choose. It can be a standalone desktop application, or a web-based system. Programming language or languages is up to your team (whatever makes sense for your project). In anticipation of questions on what type of project may qualify, here is an example: perhaps a tool for learning the fundamentals of different algorithms for a CS problem, such as sorting, where you might use a combination of animations and interactive walk-throughs where the user indicates what they think should happen. Note that no two teams may have essentially the same project (e.g., a sorting algorithm tutorial system).

Reuse of Existing Components: Your team is allowed to reuse existing open source components, provided you do the following: (a) identify the source of any components, or other code you reuse, (b) provide a copy of the license agreement for the component, code, etc, and (c) haven't violated that agreement in any way. Additionally, you may not simply "reuse" an entire open source system that happens to meet the project requirements. Additionally, as the project is to provide you with a significant development experience, your team must do a significant amount of the coding. Specifically, no more than 30% of the system's code is allowed to be existing open source components. Note that I am not counting the standard APIs of your chosen programming language toward that 30%. If you're using development tools that generate some code for you, then that auto-generated code also doesn't count toward the 30% limit on code reuse. In the unlikely event that one team develops code that is useful to another team, it may be shared at their choosing, provided that the borrowing team gives credit to the team that developed it (e.g., identifies it in their code). That code will count toward the 30% limit for the borrowing team. *Failure to give proper credit, whether to another team, or to the source of an open-source component is an academic honesty violation.* Although not an academic honesty violation, exceeding the 30% limit even if properly credited will result in a decreased project grade.

This page is intentionally blank
(except for this notice)
☺

CSIS 4485: Software and Security Engineering
Fall 2016
Team Contract

Each team must prepare this Team Contract prior to starting work on group projects. All project deliverables will receive a single grade from the instructor for the team. You will have access to the grading rubrics prior to the deadlines for deliverables. Unless your contract indicates otherwise, all members of a team will receive the same grade for each deliverable.

This contract serves two purposes: (a) to provide you with the opportunity to specify your group's preferred means of communications, to set up regular meeting schedules, team goals, roles for team members, consequences of actions (or inactions) of group members; and (b) to provide teams with a mechanism to hold themselves accountable and responsible for their actions by specifying team rules.

This contract should be developed and signed BEFORE starting the actual course project. Team membership will be finalized at the start of week two of the semester (immediately after drop-add ends). At that time, your team should meet together to develop this contract. All team members should be involved in this step, and all must come to a consensus on the details of the contract. For the Group Contract to be valid, provide the signed and dated original typed contract to me by the specified deadline (indicated in class). Give a photocopy of the original to each group member for their records.

Team Membership: List the names (printed or typed), e-mail addresses, and any other communications means that your team members may use to communicate with each other. The instructor only requires names and e-mail addresses here.

Team member name (required)	E-mail Address (required)	Other Communications Info

Team Meeting Schedule (required): Commit to the exact days/times/locations for team meetings so you don't have to negotiate this later in the semester. Specify if you meet daily, weekly, bi-weekly, the night before deliverables are due, etc? Where possible, specify exact dates of special meetings where attendance is required from all members. How long will your meetings last? Where will you meet? What other expectations are there for group meetings? What is your group's policy on attendance?

Team Action Plans: Present your group's action plan for completing the project. In essence, address how you will split the workload, yet work on this as a group in order to provide equal learning opportunities and create the best project possible. You do not have to address every bullet point below. Rather make some decisions regarding what feels right to your team. You may wish to include issues such as:

- Who will be responsible for submitting the final deliverables?
- How will you divide the work to ensure it is equitable?
- How will you record what transpired at each meeting? Will someone be designated as the group secretary or will you rotate this duty?
- How and when will you evaluate each other's work before it is placed into the final versions of deliverables?
- Will you work on all parts of the project individually and then discuss your results at group meetings; or will you do all work at group meetings?
- Will you make all of your decisions and have all discussions at the group meetings? Will decisions and discussions be conducted electronically?
- What is your proposed schedule to tackle the individual components of this project?
- How will you go about resolving group conflicts?

Attach additional sheets if necessary.

Team Roles: You are not required to specify roles here (e.g., if you envision rotating in such a way that everyone experiences different roles during the semester). However, some teams might find it useful to assign certain aspects of teamwork to certain members based on their individual strengths, etc. Some roles you may find useful to define and assign to team members include: (a) leader or manager (or whatever you want to call it)—basically someone to coordinate your team's tasks; (b) a "secretary" (or whatever you want to call it)—basically someone who is primarily in charge of keeping minutes of your team's meetings; and (c) anything else you may dream up (e.g., if you know one of your team members is the strongest programmer, then you might designate them as "lead programmer", etc). Note: If you designate roles, then you do not necessarily need every member to have such a designated specific role, and a single member can potentially have multiple roles.

Role Title	**Team Member Name**	**Responsibilities of Role**

Procedures and Penalties:
What procedures and penalties do you wish to implement in the case of "group slackers" or group members who deviate from your Team Contract? At the end of the semester, your team will have 100*N points to allocate among the N members of your team within the following constraints:

- No team member can be allocated more than 110 points.
- No team member can be allocated less than 70 points.
- The sum of all team members allocated points can be no higher than 100*N, where N is the number of team members.
- A team member's individual grade on project deliverables will be adjusted based on their team-assigned point allocation according to: (points / 100) * deliverableGrade.
- Yes, it is possible through this that some team members may receive grades greater than 100%.

Formulate the details of the procedure your team will use to determine this allocation. I suggest that you are as specific as possible to avoid later group conflict. Note: Every individual's strengths are different. E.g., someone who is a strong programmer may or may not be the best group contact person, etc. Here are some things worth considering:

- Does your team want a penalty for missing a team meeting without proper excuse?
- Does your team want to initiate a penalty for completing team-assigned tasks late?
- Is there anything your team wants to reward? Or anything else your team wants to penalize?
- How will you document any penalties or point rewards in case of later conflict?

Note that you can invent your own point system here that is separate from the one I designated above. However, if you do, make sure you also specify how you will later map your system into the point allocations I will eventually ask you for.

Make sure the details here are as explicit and clear as possible. The instructor will not directly use this. Your team will, and all members of your team will need to be in unanimous agreement over the allocation submitted to the instructor at the end of the semester. If it is not unanimous, both sides of the disagreement will need to submit to me evidence in support of their case. At which point I will turn to the details in your team's contract to resolve the dispute.

Penalty / Reward Details: (attach additional sheets if necessary)

Team Signatures: The undersigned agrees to be accountable and responsible to all of the team assigned tasks, roles, and responsibilities as set forth within this Team Contract. They further agree to the team determined procedures and penalties described in this document for allocating team grades among individual team members. Furthermore, the undersigned certify that all team members were involved in developing the details of this Team Contract.

Team Member Name (print)	Team Member Signature	Date

This page is intentionally blank
(except for this notice)
☺

CSIS 4485: Software and Security Engineering
Fall 2016
End of Semester Team Point Allocation
Deadline: December 8, in class, on paper, and signed by the team.

Overview:
Your team has 100*N points to allocate among the N members of your team within the following constraints:

- No team member can be allocated more than 110 points.
- No team member can be allocated less than 70 points.
- The sum of all team members allocated points can be no higher than 100*N, where N is the number of team members.
- A team member's individual grade on project deliverables will be adjusted based on their team-assigned point allocation according to: (points / 100) * deliverableGrade.
- Yes, it is possible through this that some team members may receive grades greater than 100%.

Your allocation should be consistent with the procedures and penalties your team set forth in your Team Contract at the beginning of the semester. Your team must have unanimous agreement over the allocation. If any individual member disagrees, then both sides will be required to submit at the time this allocation is submitted documentation in support of their side. I recommend discussing this with your team prior to December 10 (perhaps near the end of the week prior). It will be easiest for your team if you are able to self-resolve any team conflicts.

Note: If all team members are in agreement, then I do not need (and do not want) the details of how you arrived at the following allocation. I just want the final allocation. I prefer whole numbers, but if you want to give fractions of points to each other that is fine as well.

Team Member Name (print or type)	**Point Allocation (of majority)**	**Alternate Point Allocation (use only if point allocation is in dispute)**
Total (must be no greater than 100*N):		

Team Signatures: The undersigned agrees that they are in agreement with either the majority point allocation or the alternate as indicated in the appropriate column in the table below. Note: If there is not complete agreement over the allocation, both those in the majority as well as anyone disputing, must provide the instructor with documentation in support of their view by the deadline for this form.

Team Member Name	**Signature**	**Agree with Majority Allocation?**	**Agree with Alternate allocation?**	**Date**