Program Outcome:

b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

CMPS 490 Computer Projects is a required course for all Computer Science students. It is taken in the first semester of senior year. Students develop and document a significant software system during this course. This documentation includes producing a graded Requirements Specification. This assignment presents an excellent opportunity to assess outcome b.

Assessment will take the form of a rubric used by the instructor when grading the Requirements Specification document. The instructor will then average results for the class as a whole and return the following summary document to the departmental assessment committee. Target scores are 66.6% for every item.

	CIS AVERAGE (n=5)	CS AVERAGE (n=11)	IT AVERAGE (n=0)
Rubric criterion			
Problem definition	1.8	2.25	NA
Requirements clarity/detail	2.2	2.4	NA
Requirements appropriateness	2.2	2.4	NA

	Level of Achievement		
	1 **Below expectations**	**2** **Meets expectations**	**3** **Exceeds expectations**
Problem definition	It is not totally clear what problem is to be solved. The problem may be poorly described.	Problem is described in a reasonable fashion. Sufficient context is provided.	Problem definition is exceedingly clear. Extensive background information is provided.
Requirements clarity/detail	Requirements are ambiguous or otherwise difficult to understand. They may be insufficiently detailed or weakly explained.	Requirements are clear and concise. The level of detail is sufficient.	Requirements have been carefully considered. They are highly detailed, but presented clearly.
Requirements appropriateness	Requirements may not be entirely appropriate. They may not reflect the entirety of the problem or may lead to impractical solutions.	Requirements represent a reasonable match between the problem and computing solutions.	Requirements are carefully constructed to offer a complete problem solution that is well-matched to computing capabilities.

Program Outcome:

c. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

CMPS 490 Computer Projects is a required course for all Computer Science students. It is taken in the first semester of senior year. Students develop and document a significant software system during this course. As part of this course, students will make a final oral presentation and demonstrate his or her project. This presentation presents an opportunity to assess outcome c.

Assessment will take the form of a rubric used by the instructor when evaluating the presentation. The instructor will then average results for the class as a whole and return the following summary document to the departmental assessment committee. Target scores are 66.6% for every item.

	CIS AVERAGE (n=5)	CS AVERAGE (n=11)	IT AVERAGE (n=0)
Rubric criterion			
Design	2.2	2.35	NA
Software implementation	2.2	2.4	NA
Testing	2.2	2.5	NA

	Level of Achievement		
	1 **Below expectations**	**2** **Meets expectations**	**3** **Exceeds expectations**
Design	The project suffers from poor overall design. It may be hampered by inappropriate module use or redundancy between modules.	Modularization is appropriate. Little redundancy exists between modules. External resources are used effectively.	Modularization minimizes code and enhances understanding. Excellent use is made of external resources. Foresight is evident.
Software implementation	Poor software implementation. Code is poorly commented and/or modularized. It may contain bugs.	Code contains reasonable comments. Appropriate use is made of libraries and modularization. Code is reliable.	Code is well-documented and organized. It is efficient and clear. Modification would be easy. Code is very reliable.
Testing	Project has not been extensively tested, or test coverage was inadequate.	Project has undergone a reasonable amount of testing. Appropriate test cases were used. Test coverage was satisfactory.	Testing was extensive and carefully constructed.

Program Outcome:

d. An ability to function effectively on teams to accomplish a common goal.

CMPS 352 Operating Systems is a required course for all Computer Science students, and it is normally taken in the first semester of their junior year. Students in the course are divided into teams with 3 or 4 students in each, and each team carries out a semester-long team project, in which it builds a client-server application using sockets, shared memory and semaphores. For this reason, we decide to use this course as a candidate for assessing our student teamwork outcome.

Assessment tools include two rubrics, *student rubric* and *instructor rubric*. The student rubric is for each student to assess his/her team, and the instructor rubric for the instructor to assess each team. Students are introduced to the two rubrics at the beginning of the semester, and they complete the student rubric by the end of the semester. The instructor completes the instructor rubric one for each team.

After collecting completed student rubrics, the instructor(s) calculates the average score of each team on each rubric criterion, and then the average of teams' average scores on each criterion. The following table is submitted to the departmental ABET Assessment Committee.

Fall 2016

Summary of Instructor Rubrics

	CS AVERAGE (n=9)	CiS AVERAGE (n=1)	SE AVERAGE (n=1)
Members commit to the team's CVS repository equally	75%	75%	50%
Each member participates in team presentation and demonstration	92%	100%	100%
Team members show courtesy and respect for each other	100%	100%	100%

Summary of Student Rubrics

	CS AVERAGE (n=9)	CiS AVERAGE (n=1)	SE AVERAGE (n=1)
Team members communicate and coordinate with each other	92%	100%	100%
Each team member does a fair share of the work at every stage of the project	78%	75%	100%
Team members actively listen to each other's ideas and collectively make team decisions	92%	100%	100%
Team members show courtesy and respect for each other	94%	100%	100%

Students' Team Assessment Rubric

Team Name: **Student Name:** **Major: CS – CIS – CE – SE -- other**

	Unsatisfactory **1 (25%)**	**Developing** **2 (50%)**	**Satisfactory** **3 (75%)**	**Exemplary** **4 (100%)**
Team members communicate and coordinate with each other	Team members never talk with each other and no one knows what other members are doing	Very few meetings (in person or via electronic means) and each member vaguely knows what other members are doing.	Members communicate and know what other members are doing, but there is no or little coordination among them.	Members are all informed of every aspect of the project and coordinate with each on tasks and progress
Each team member does a fair share of the work at every stage of the project	One or two students do all the team work or some members do not contribute at all	Some members contribute significantly more than others or some members contribute significantly less	Some members do more than others at certain stages, but overall, all members fairly share team work	All members consistently fairly share team work in every stage of the project
Team members actively listen to each other's ideas and collectively make team decisions	One member of the team talks all the time and makes all the decision, or no one actively participates in any meeting and no decisions are made.	One member dominates the talks, but sometimes other members provide help when requested	Members generally listen to each other and collectively make at all major decisions	Members always listen to each other and make collectively decisions.
Team members show courtesy and respect for each other	Members are rude and disrespectful to each other	Some members may be discourteous to others occasionally.	Most members show courtesy and respect to each other almost all the time.	All members are always courtesy to and respectful to each other

Instructor's Team Assessment Rubric

Team Name:

	Unsatisfactory **1 (25%)**	**Developing** **2(50%)**	**Satisfactory** **3 (75%)**	**Exemplary** **4(100%**
Members commit to the team's CVS repository equally	All commits are made by one member	A significant numbers of commits are made by a single member and some member(s) has no commits	All members have commits, but some members have commits than others.	All members have about the amount of commits
Each member participates in team presentation and demonstration	One member does the presentations and demos all the times	One member delivers the presentation and demo, and other members offer some comments	All members participate, but different members contribute differently	All members always work together to make coherent and collaborative presentations and demonstrations.
Team members show courtesy and respect for each other	Members complain about and/or discredit each other, and/or show disrespect to each other, for instance, members always interrupt each other in meetings	Members show no visible disrespect to each other, but there is no communication and collaboration.	Members are generally courteous and respectful to each other, and they work together, they have to be reminded to work as a team.	Members are always courteous and respectful to each other.

Program Outcome:

f. An ability to communicate effectively with a range of audiences.

CMPS 490 Computer Projects is a required course for all Computer Science students. It is taken in the first semester of senior year. It is a capstone project course that contains a significant communication component. Projects are presented orally to the class. Additionally, substantial written documentation is produced. The writing component qualifies the course for a writing-intensive designation under the University's general education system. As such, this course presents an ideal opportunity to assess outcome f.

Assessment tools include two rubrics, *oral communication* and *written communication*. The instructor completes both rubrics. The instructor then averages the rubric results for the class and compiles the following table to be submitted to the departmental assessment committee. Target scores are 66.6% for every item.

	CIS AVERAGE (n=5)	CS AVERAGE (n=11)	IT AVERAGE (n=0)
Rubric criterion			
Organization (oral)	2.2	2.5	NA
Mechanics (oral)	1.8	2.5	NA
Delivery (oral)	1.8	2.6	NA
Relating to audience (oral)	2.2	2.6	NA
Style (written)	1.8	2.4	NA
Organization (written)	2.2	2.6	NA
Visual elements (written)	2.2	2.55	NA

Oral Communication

	Level of Achievement		
	1 **Below expectations**	**2** **Meets expectations**	**3** **Exceeds expectations**
Organization	Audience has difficulty following presentation because of some abrupt jumps; some of the main points and conclusion are unclear.	Satisfactory organization; clear introduction; main points are well stated, even if some transitions are somewhat sudden; clear conclusion.	Superb organization; clear introduction; main points well stated and argued, with each leading to the next point of the talk; clear summary and conclusion.
Mechanics	Boring slides; numerous mistakes; no real effort made into creating a truly effective presentation; poor participation of team members.	Generally good set of slides; conveys the main points well. Adequate participation of team members.	Very creative slides; carefully thought out to bring out both the main points as well as the subtle issues while keeping the audience interested. Excellent participation of team members.
Delivery	Low voice, occasionally inaudible; some distracting filler words and gestures; pronunciation not always clear.	Clear voice, generally effective delivery; minimal distracting gestures, but somewhat monotone.	Natural, confident delivery that does not just convey the message but enhances it; excellent use of volume and

			pace.
Relating to audience	Occasional eye contact with audience but mostly reads the presentation; some awareness of a least a portion of the audience; only brief responses to questions.	Generally aware of the audience reactions; maintains good eye contact when speaking and when answering questions.	Keeps the audience engaged throughout the presentation; modifies material on-the-fly based on audience questions and comments; keenly aware of audience reactions.

Written Communication

	Level of Achievement		
	1 **Below expectations**	**2** **Meets expectations**	**3** **Exceeds expectations**
Style	Text rambles, key points are not organized; spelling or grammar errors present throughout more than 1/3 of paper; style is inappropriate for audience; prescribed format is not followed.	Articulates ideas; one or two grammar or spelling errors per page; prescribed format is followed.	Articulates ideas clearly and concisely; presented neatly and professionally; grammar and spelling are correct; uses good professional style; and conforms to prescribed format.
Organization	Material generally well organized, but paragraphs combine multiple thoughts or section / subsections are not identified clearly.	Generally organizes material in a logical sequence, but some material may be out of place.	Organizes material in a logical sequence to enhance reader's comprehension (paragraph structure, subheadings, etc.).
Visual elements	Uses images, tables, diagrams, but only in a few instances are they used to support, explain, or interpret information; visual elements may contain	Some use of images, tables, diagrams to support points; to explain, interpret, and assess information; figures are all in	Uses images, tables, diagrams to support points; to explain, interpret, and assess information; figures are all in

	flaws	proper format.	proper format.

Rubric adapted from http://www.eng.mu.edu/corlissg/advice/communications_rubrics.html

Program Outcome:

h. Recognition of the need for and an ability to engage in continuing professional development

The University of Scranton ACM chapter is an organization on campus that is closely tied with professional engagement and development for computing students. In particular, it arranges lectures by high-profile external speakers and trips to computing conferences and employers. Various metrics of involvement in these activities may serve as an indicator for outcome h.

Quantitative assessment would include the following:

	CIS Average (N=12)	CS Average (N=58)	IT Average (N=6)
Metric			
Attendance at annual ACM distinguished speaker presentation (target 50%)	16.6%	36.2%	0%
Number of students that are members of national ACM organization (target 20%)	0%	15.5%	0%

Note: DSP data may be biased this year due to use of extra credit incentives.

Assessment of program outcomes I, J, K in CMPS490

CMPS490 Senior Projects is a required course for all Computer Science students, and it is normally taken in the first semester of their senior year. Each student works with a faculty member as his/her adviser.

Assessment tools include a rubric for each student to be completed by the student adviser. The course instructor collects all the rubrics from advisers, completes the summary table, and submits it to the departmental ABET Assessment Committee.

(i) An ability to use current techniques, skills, and tools necessary for computing practice.

To be completed by the instructor of the course

Summary of Course Rubrics

	CIS AVERAGE (n=5)	**CS AVERAGE (n=11)**	**IT AVERAGE (n=0)**
Use of toolkits and frameworks	3.333333333	3.111111111	NA
Programming language	3.333333333	3.222222222	NA
Use of software development tools	3.333333333	3.222222222	NA
Use of documentation tools	3.333333333	2.722222222	NA

Student Name: **Adviser Name:**

	Unsatisfactory 1	**Developing 2**	**Satisfactory 3**	**Exemplary 4**
Use of toolkits and frameworks	Has no knowledge of toolkits/ frameworks relevant to the project and/or no rationale on the choice of	Uses Proper toolkits/ frameworks suitable for the project, but needs much	Uses some toolkits/ frameworks suitable for the project with major	Has excellent knowledge of toolkits/ frameworks suitable for the project and uses most

	toolkits/frameworks chosen.	assistance	errors/mistakes	suitable ones with little errors/mistakes.
Programming language	Knows only one or few programming languages that are suitable for the project and/or provides no rationale for the language(s) chosen for the project	Chooses a language(s) suitable for the project, but needs much assistance in using the language(s)	Chooses a language(s) suitable for the project and uses it effectively with no errors/mistakes	Has excellent knowledge of programming languages suitable for the project and uses the language(s) effectively and efficiently with no errors/mistakes (e.g, composes efficient queries, uses batch insert when possible in SQL)
Use of software development tools	Knows no or very few tools and/or does not use proper tools for the system	Knows some tools and uses some in the development process	Knows proper tools for the components of the systems, does use appropriate tools for the jobs, but not to the fullest potential of the tools	Has excellent knowledge of tools for the system development and use all the right tools to their fullest potentials
Use of documentation tools	Uses no proper tools used for the whole or part of the document, e.g., hand-drawn diagrams.	Uses some documentation tools for part of the system (e.g., table of contents and indexes composed manually in MS Word, parameters not specified in JavaDoc)	Uses proper tools for most of the system effectively with no errors/mistakes	Uses most effective documentation tools effectively for different part of the system with no errors/mistakes

(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]

To be completed by the instructor of the course

Summary of Course Rubrics

	CIS AVERAGE	**CS AVERAGE**	**IT AVERAGE**
Mathematical	3.333333333	3.166666667	NA

models and functions			
Algorithmic principles	3.333333333	3.388888889	NA
Data structures and file management	3.333333333	3.277777778	NA
Database	3	3.111111111	NA

Student Name: **Adviser Name:**

	Unsatisfactory **1**	**Developing** **2**	**Satisfactory** **3**	**Exemplary** **4**
Mathematical models and functions	Has numerous mathematical errors and mistakes in modeling and/or programs, or uses incorrect mathematical functions/ expressions	Uses correct mathematical models and functions, but needs significant assistance	Uses correct mathematical models and functions correctly with no major mistakes	Uses correct mathematical models and functions correctly with no errors/mistakes independently
Algorithmic principles	Has numerous algorithmic errors and mistakes such as Boolean conditions not considered or implements algorithms incorrectly, or uses incorrect algorithms.	Uses correct algorithms for different problems, but needs significant help in understanding or implementing the algorithms.	Chooses suitable algorithms for different problems and implements them with no errors/ mistakes, maybe need some help on some algorithms	Chooses s suitable algorithm for each problem with correct implementation of the algorithm in the given programming language independently
Data structures	Gives no consideration in data structure design and tradeoff, or uses data structures incorrectly	Chooses data structures for the problem, but needs much help in choosing and/or implementing data structures or makes numerous errors/ mistakes	Chooses data structures suitable for the problem and implements them with no or few errors/ mistakes.	Studies different design alternatives and chooses data structures most suitable for the problem. Implements the data structures and/or file organization with no

				errors/mistakes
File organization and Database	Gives no consideration in use of files and database or in design of the database and/or file organization, or designs/ implements the database or file organization with major error/ mistakes.	Chooses databases or file organization suitable for the system, but needs great assistance in their design and/or implementation, or makes major errors or mistakes	Designs and implements file organization and/or databases suitable for the application with no major errors/mistakes	Studies different design alternatives and chooses database models and DBMS or file organization most suitable for the system. Designs and implements the database and/or file organization with no or little errors/ mistakes

(k) An ability to apply design and development principles in the construction of software systems of varying complexity.

To be completed by the instructor of the course.

Summary of Course Rubrics

	CIS AVERAGE	**CS AVERAGE**	**IT AVERAGE**
Development Process	3	3.333333333	NA
Software Architecture	3.333333333	3.555555556	NA
Coding Styles and Standards	3	2.833333333	NA
Testing	3	2.9444	NA
Deployment	1.333333333	3.111111111	NA
Documentation	3.333333333	2.666666667	NA

Student Name: **Adviser Name:**

	Unsatisfactory 1	**Developing 2**	**Satisfactory 3**	**Exemplary 4**
Development process	Does not know or have any knowledge	Understands the basic phases of	Understands and adopts a	Knows a few different development process

	of any development process models	software development cycle and tries to follow the basic requirements, designs, coding, testing steps, but does not know exactly what to do at each step.	development process model for the development, but does not follow the model consistently, which causes necessary repetition and modification of design and coding	models and consciously choose a model suitable for the application, and follows the model entirely for the development.
Software architecture	Gives no consideration in system decomposition and architecture	Decomposes the system into logical modules based on some domain knowledge, but presents no real architecture of the system	With some assistance, decomposes the system into logical modules using software design principles based on clear domain knowledge, and presents a software architecture suitable for the application	Independently, composes the system into logical modules using software design principles, and presents a clear software architecture with no or little errors or mistakes
Coding styles and standard	Follows no coding standards with no or few comments in the code, gives no effort in using meaningful names for various entities.	Shows some effort in commenting the code and in naming variables and operations, but does not show a consistent effort in doing so	Provides comments for most of major functions and gives major functions or variable meaningful names, but not consistently throughout the whole system	Provides comments for all the modules/ functions, consistently give functions and variables meaningful names, uses a standard (such as Javadoc) in documenting the code
Testing	No unit and/or system testing are performed and no test plan is presented.	Random unit testing and/or system testing are performed and a test plan is presented, but has errors and is not complete	Unit testing and system testing are performed with some flaws and test plan is not all complete	Unit testing and system testing are all performed with few errors and test plan is complete and well written
Deployment	No deployment plan and no deployment	Deployment plan is presented, but it is not complete and/or	Deployment plan is presented, and it is carried out with no	Deployment plan is clear and complete, it is completed with no

		not coherent and/or deployment was not successful	major errors	errors
Documentation	No or little documentation	Documentation is not complete and logically coherent	Documentation is complete, but has a few minor errors	Documentation is complete and well formatted with few or no errors