COMPUTER SCIENCE ASSESSMENT PLAN
School of Information Technology

Program Educational Objectives:

The program educational objectives (PEO) of the computer science program are as follows:

1. Be a successful practitioner in a computer science related field or accepted into a graduate program.
2. Design and develop creative and effective solutions to practical computing problems.
3. Exhibit teamwork and effective communication skills.
4. Be characterized by effective leadership skills and high standards of ethics.
5. Engage in lifelong learning to adapt to an ever-changing professional environment.

Student Outcomes:

At the time of graduation, a student in our computer science program must attain the following outcomes:

a. An ability to apply knowledge of computing and mathematics appropriate to the discipline
b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
c. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired standards
d. An ability to function effectively on teams to accomplish a common goal
e. An understanding of professional, ethical, legal, security and social issues and responsibilities
f. An ability to communicate effectively with a wide range of audience
g. An ability to analyze the local and global impact of computing on individuals, organizations, and society
h. Recognition of the need for and an ability to engage in continuing professional development
i. An ability to use current techniques, skills, and tools necessary for computing practice
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
k. An ability to apply design and development principles in the construction of software systems of varying complexity

Relationship of Student Outcomes to Program Educational Objectives

The table below summarizes the relationship between student outcomes and program educational objectives:

Student Outcomes	Program Educational Objectives				
	1	2	3	4	5
(a)	▪	▪			
(b)	▪	▪			
(c)	▪	▪			
(d)	▪		▪		
(e)	▪			▪	
(f)	▪			▪	
(g)	▪			▪	
(h)	▪				▪
(i)	▪	▪			
(j)	▪	▪		▪	
(k)	▪	▪		▪	

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Expected level of attainment ***	**Timeline**
Measures computing performance	IT 179, IT 225, IT 279, IT 328, IT 356, IT 378, IT 383	IT 279, IT 383	IT 279: use first two rows of rubric (a)(i) IT 383: use third row of rubric (a)(i)	IT 279: Homework or paper that involves efficiency analysis IT 383: Assignment(s) that deal with understanding and analyzing computing performance	IT 279 students IT 383 students	70% (IT 279) 70% (IT 383)	IT 279: Even fall semesters IT 383: Odd Spring semesters
Uses number systems and Boolean logic to describe how computing systems work	IT 225, IT 226, IT 279	IT 225	Use rubric (a)(ii)	IT 225: Assignment(s) that deal with primitive data types and use of Boolean logic to design logic circuits	IT 225 students	50%	Odd Spring semesters
Writes computer programs that use mathematical results	IT 179, IT 279, IT 328, IT 356	IT 279	Use rubric (a)(iii)	IT 279: Assignment(s) that use or implement mathematical concepts	IT 279 students	70%	IT 279: Even fall semesters IT 168: Odd spring semesters
Describes how concepts of computability are related to practical computing	IT 328	IT 328	Use rubric (a)(iv)	IT 328: Assignment(s) relating concepts of computability to practical computing	IT 328 students	70%	Data collected each semester; reviewed each fall for the previous year

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

Rubric (a)(i)				
	Poor or Non-Existent	Developing	Developed	Exemplary
Understands the use of big-O notation in comparing competing algorithms	Does not compare algorithms using big-O notation, but other unreliable methods	Uses big-O notation correctly only in a few cases to compare algorithms, but mostly uses other unreliable methods	Uses big-O notation in nearly all cases to correctly compare and contrast competing algorithms	Uses big-O notation in all cases to compare algorithms, while correctly identifying its limitations
Demonstrates an understanding of how theoretical efficiency of an algorithm is related with actual timing of its implementation	Does not show any understanding of how big-O complexity and actual timing of an implemented algorithm are related.	Understands big-O notation but not able to explain its relationship (supporting or contradicting) with actual timing of an implementation	Explains how theoretical complexity of an algorithm and actual timing of its implementation correlate to each other, or why they seem to contradict each other.	Explains in detail how actual timing of an implemented algorithm supports or contradicts its theoretical complexity by showing the limitations of each approach.
Demonstrates an understanding of the major factors determining the computing performance and how to analyze the performance	Does not show any understanding of the major hardware and software factors determining computing performance and how to analyze the performance using the given performance parameters	Explains the major hardware and software factors determining computing performance and how to analyze the performance using the given performance parameters, but does not demonstrate how to analyze performance using given performance parameters	Explains the major hardware and software factors determining computing performance and how to analyze performance using the given performance parameters (e.g., cache hit rate, TLB miss rate, structure of page table, disk access time, and multithreaded programming)	Explains the major hardware and software factors determining computing performance and how to analyze the performance using the given performance parameters and also explain s the benefits and limitations of different approaches to enhance computing performance

Rubric (a)(ii)				
	Poor or Non-Existent	Developing	Developed	Exemplary
Understand primitive data types (e.g., integer, floating points, and characters) are represented in binary numbers	Does not explain how to represent most of primitive data types in binary numbers	Explains how to represent most of primitive data types in binary numbers	Explains how to all primitive data types in binary numbers: -use 2's (1's) complement representation to express integers - use ASCII code to represent characters -use IEEE-754 format to represent floating points	Explains how to represent all primitive data types using binary numbers and also explains the limitations and benefits of each data representation
Explains how Boolean logic can be used to design various combinational and sequential logic circuits	Does not show any understanding of how Boolean logic can be used to implement combinational logic or sequential logic circuits	Understands how to design combinational logic using sum-of-product notations but not able to explain how to design sequential logic	Explains how Boolean logic can be used to design the following combinational and sequential logic circuits - Full adder - ALU - Multiplexor - decoder - latch and flip-flops	Explains how Boolean logic can be used to design major combinational and sequential logic circuits and also explain how to interpret timing diagram

Rubric (a)(iii)				
	Poor or Non-Existent	Developing	Developed	Exemplary
Writes computer programs that use mathematical results	Program does not work, and student cannot relate the math concept with the programming task	Program works only in some cases, or works only partially	Program works correctly in all specified cases and implements the math as specified	Program works correctly and student addresses special cases not directly specified in instructions

Rubric (a)(iv)				
	Poor or Non-Existent	Developing	Developed	Exemplary
Understand the concepts of deterministic and nondeterministic model	Does not know the meaning and difference between the two models	Knows the difference and how to use the nondeterministic model to describe a few problems	Knows the difference and how to use the nondeterministic model to describe all typical problems	Knows the difference and how to use the nondeterministic model to describe all typical typical problems, and the complexity and computability implications of the nondeterministic model under different constraints
Understand Regular Languages	Does not know what are they	Knows at least two of the three formalisms to describe Regular Languages, i.e., Regular expressions, Regular Grammars, and Finite State Automata	Knows all of the three formalisms and know how to convert from one to another	Knows how to optimize Finite State Machines (from NFA to DFA and minimize the state)
Understand Context-free Languages	Does not know what are they	Knows Context-free Grammars and Pushdown Automata	Knows how to convert between Context-free grammar and Pushdown Automata	Identifies some ambiguous Context-free languages and the limitation of deterministic Pushdown Automata
Understand Turing Machines	Does not know what are they	Knows the definition, components, and operations of Turing machines	Knows the concept of computability and Universal Turing Machines	Knows the limitation of Turing Machines and be able to identify a few undecidable problems
Understand the concepts of NP-Completeness	Does not know P vs NP	Knows the definitions of P, NP and NPC	Knows how to prove some problems being NPC	Knows a few approaches to prove or disprove NP=P

(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Expected level of attainment***	**Timeline**
Uses common modeling techniques to analyze a problem	IT 261, IT 326, IT 378,	IT 261	Use rubric (b)(i)	IT 261: Assignment(s) that deal with using models to analyze a problem	IT 261 students	50%	Even Fall semesters
Gathers requirements for a given problem	IT 261, IT 378	IT 261	Use rubric (b)(ii)	IT 261: Assignment(s) that deal with gathering requirements for a given problem	IT 261 students	50%	Even Fall semesters

Rubric (b)(i)				
	Poor or Non-Existent	Developing	Developed	Exemplary
(i) Use common modeling techniques to analyze a problem	Unable to produce recognizable model	Can create visual model, but model does not fit problem	Creates visual model that fits problem description	Creates a well-formed and parsimonious model of problem
(ii) Perform requirements gathering	Records none or very few requirements	Record some appropriate requirements but misses one or more major requirements	Records all appropriate requirements	Records all appropriate requirements in a well-formatted and logical manner

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Expected level of attainment***	**Timeline**
Uses common modeling techniques to design a solution	IT 261, IT 226, IT 326, IT 378,	IT 226	Use rubric (c)(i)	IT 226: Assignment(s) that deal with using models to design a solution to a given problem	IT 226 students	60%	Odd Fall semesters
Writes a computer program that solves a problem	IT 168, IT 179, IT 226, IT 279, IT 328, IT 340, IT 353, IT 356, IT 383	IT 168, IT 279	Use rubric (c)(ii)	IT 168 and IT 279: Programming assignment(s)	IT 168 students IT 279 students	50% (IT 168) 70% (IT 279)	IT 168: Odd Spring semesters IT 279: Even fall semesters
Evaluates alternative solutions for a given problem	IT 279	IT 279, Exit Exam	Use rubric (c)(iii)	IT 279: Homework or paper that deals with proposing or evaluating multiple solutions to the same problem Exit Exam: Question(s) relating to evaluating alternative solutions to a given problem	IT 279 students Students taking exit exam	70% (IT 279) 70% (Exit Exam)	IT 279: Even fall semesters Exit exam: Even fall semesters, Odd Fall semesters
Writes test cases for a problem and/or implementation	IT 226, IT 326	IT 226	Use rubric (c)(iv)	IT 226: Assignment(s) that ask students to write test cases for a given problem or implementation	IT 226 students	60%	Odd Fall semesters

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

Rubric (c)(i)

	Poor or Non-Existent	Developing	Developed	Exemplary
Use common modeling techniques to design a solution	Unable to create a recognizable model	Create models but models do not fully represents the problem domain or are not consistent with the specified modeling language	Create models that represent the problem domain and are consistent with the specified modeling language	Creates a well-formed and parsimonious design model that can be used by an external coder for developing a computer application

Rubric (c)(ii)

	Poor or Non-Existent	Developing	Developed	Exemplary
Writes computer program that solves a problem	Program has major syntactical errors or does not run with normal inputs without crashing, code does not solve the given problem	Program produces correct results in only some cases, program crashes with some valid inputs	Program works correctly for all sample data and typical cases, solves the correct problem	Program works correctly for all relevant cases, and addresses at least one unspecified case or implements an extra feature

Rubric (c)(iii)

	Poor or Non-Existent	Developing	Developed	Exemplary
Evaluates alternative solutions for a given problem	Student does not correctly identify at least two correct solutions for the given problem, does not use correct methods to evaluate them	Student identifies correct alternatives but evaluates them incorrectly or fails to include evaluation	Student identifies correct alternatives, uses correct evaluation methods and reaches correct conclusions	Student goes beyond requirements, presents detailed and correct evaluation of each alternative solution

Rubric (c)(iv)

	Poor or Non-Existent	Developing	Developed	Exemplary
Writes test cases for a problem and/or implementation	Student does not write meaningful test cases for the problem	Student writes cases that mostly test only superficially	Student writes meaningful test cases that sufficiently test for a problem/implementation	Student writes at least one test case that meaningfully tests the problem and was not identified by the instructor

(d) An ability to function effectively on teams to accomplish a common goal							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Expected level of attainment***	**Timeline**
Contributes fairly to tasks assigned to the team	IT 226, IT 378, IT 391, Internship survey	IT 226	Use rubric (d)(i)	IT 226: Peer and group reviews from group assignment(s) or projects	IT 226 students	60%	Odd Fall semesters
Actively participates in team discussions							
Describes all team roles and responsibilities							

Rubric (d)(i)				
	Poor or Non-Existent	Developing	Developed	Exemplary
Contributes fairly to tasks assigned to the team	Does not contribute to final deliverable	Completes assigned tasks only partially	Satisfactorily completes assigned parts	Completes assigned parts and helps other team members with their assigned work, initiates and participates in team meetings
Actively participates in team discussions	Does not contribute to discussions, does not let others express opinions	Contributes occasionally to team discussions	Contributes equally in team discussions	Leads team discussions, ensures that everybody is heard
Describes all team roles and responsibilities	Does not know what any other team member is doing	Knows only what some team members are doing, and not others	Describes clearly the role and responsibility of each team member	Motivates others to fulfill their responsibilities

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

(e) An understanding of professional, ethical, legal, security and social issues and responsibilities

Performance Indicator	Delivery Methods	Courses used for Assessment	Assessment Methods	Data Needed	Assessed Groups	Expected level of attainment**	Timeline
Identifies security considerations for IT systems	IT 179, IT 350, IT 378	Exit Exam	Use rubric (e)(i)	Exit Exam: Question(s) relevant to identifying security considerations for IT systems	Students taking Exit Exam	70%	Data collected each semester; reviewed each fall for the previous year
Identifies proper use of existing software programs based on licenses	IT 226, IT 356	Exit Exam	Use rubric (e)(i)	Exit Exam: Question(s) relevant to identifying whether existing software programs can be used in a specific setting based on their licenses	Students taking Exit Exam	70%	Data collected each semester; reviewed each fall for the previous year
Identifies sections of a professional code of ethics that apply to a given situation	IT 191, IT 279	Exit Exam	Use rubric (e)(i)	Exit Exam: Question(s) that relate sections of a professional code of ethics to a given situation	Students taking Exit Exam	70%	Data collected each semester; reviewed each fall for the previous year

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

Rubric (e)(i)				
	Poor or Non-Existent	Developing	Developed	Exemplary
Identify security considerations for IT systems	Cannot identify security considerations	Identifies a few security considerations	Identifies security considerations appropriate for system	Identifies security considerations appropriate for system. Ranks considerations according to risk and provides reasoning for ranking.
Identifies sections of a professional code of ethics that apply to a given situation	Does not identify sections of a relevant professional code of ethics	Identifies sections of a code of ethics, but some are not relevant to the situation and/or some relevant sections are not identified	Identifies the sections of a code of ethics that are relevant to the situation with at most one irrelevant section or one relevant section not identified	Identifies exactly the relevant sections and applies them appropriately to the situation

(f) An ability to communicate effectively with a range of audiences

Performance Indicator	Delivery Methods	Courses used for Assessment	Assessment Methods	Data Needed	Assessed Groups	Expected level of attainment*	Timeline
Communicates effectively with a range of audiences orally	IT 191, IT 226, IT 353, IT 378, IT 391, COM 110	IT 226	Use rubric (f)(i)	IT 226: Oral Presentation	IT 226 students	60%	Odd Fall semesters
Communicates effectively with a range of audiences in writing	IT 191, IT 226, IT 279, IT 327, IT 328, IT 378, IT 391, ENG 101, ENG 249	IT 279	Use rubric (f)(ii)	IT 279: Written paper	IT 279 students	70%	Even Spring semesters

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

Rubric (f)(i)

	Poor or Non-Existent	Developing	Developed	Exemplary
Clarity	Not assertive or clear overall	Assertive but inconsistent, occasionally trying to sound too technical or intentionally vague	Mostly clear and easy to understand	Clear and assertive, very easy to understand
Organization	Not well organized, no logical flow	Inconsistent flow, lacking macro or micro organization	Logically organized at micro and macro level	Entire communication has logical flow, flow is reinforced throughout
Audience	Not aimed at the intended audience	Reflects own knowledge rather than targeting audience, could have taken more efforts to direct talk at audience	Directed at appropriate audience	Targeting audience well enough to enhance communication
Engaging the audience	Not captivating, could not engage audience, little to no interaction with audience	Good beginning and end but not as engaging in between, not enough interaction with audience	Keeps the audience interested and facilitates some interaction	Keeps the audience awake and involved, occasionally adapting to audience's feedback
Delivery	Two or more of: Spoke too fast/too slow, did not address intended questions, inappropriate attire, took significantly longer or shorter than allotted time	One of: Spoke too fast/too slow, too many pauses, awkward body language	Spoke at appropriate pace, comfortable and appropriate body language	Calm. Clear diction. Good tone. Good pacing. Appropriate attire and personal grooming.

Rubric (f)(ii)

Written Communication

	Poor or Non-Existent	Developing	Developed	Exemplary
Clarity/ Precision	Too vague or too detailed, significant amount of information may be inaccurate.	Detailed but losing overall picture, or clear at a high level but missing details, attention to length rather than substance. Some information may be inaccurate.	Appropriately detailed and focused at a higher level. Writing is precise and concise.	Completely clear and precise
Organization	Not well-organized, no consistent flow	Micro-structure well defined but lacking macro-structure, or vice versa	Good and appropriate organization	Logically organized
Audience	Not catered to intended audience (wrong assumptions about audience, trying to target all types of audiences)	Not consistently aiming at the audience, occasionally too detailed or too vague	Most aiming at the appropriate audience	Aimed exactly at the appropriate audience
Mechanics and Style	Many spelling and grammar errors, no logical flow or document structure	Logical flow but with many spelling and grammar errors, or vice versa, crude document structure	No spelling or grammar errors. Reasonably good logical flow and appropriate document structure	No spelling or grammar errors. Good use of language and good logical flow
Visual aids	No visual aids/too many visual aids. Very poor visual aids.	Few visual aids, some incompletely made, not referred in the text. Some visual aids poorly designed	Appropriate number and kind of visual aids referred by the text at the proper places parts	Appropriate number of well-chosen visual aids that enhance communication

(g) An ability to analyze the local and global impact of computing on individuals, organizations, and society							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Expected level of attainment***	**Timeline**
Analyze the local and global impact of software programs and its applications on individuals, organizations and society	IT 226, IT 326, IT 340, IT 378	Exit Exam	Use rubric (g)(i)	Exit Exam: Question(s) that are relevant to the impact of computing on individuals, organizations and society	Students taking Exit Exam	70%	Data collected every semester; reviewed each fall for the previous year

Rubric (g)(i)				
	Poor or non-existent	Developing	Developed	Exemplary
Ability to analyze the local and global impact of computing on *individuals*	Test score <70	Test score between 70 and 80	Test score between 80 and 90	Test score above 90
Ability to analyze the local and global impact of computing on *organizations*	Test score <70	Test score between 70 and 80	Test score between 80 and 90	Test score above 90
Ability to analyze the local and global impact of computing on *society*	Test score <70	Test score between 70 and 80	Test score between 80 and 90	Test score above 90

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

(h) Recognition of the need for and an ability to engage in continuing professional development							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Expected level of attainment***	**Timeline**
Participates in independent learning	IT 279, independent studies	IT 279	Use rubric (h)(i)	IT 279: Paper that involves student work on topics not included in syllabus	IT 279 students	70%	Even Spring semesters
Participates in IT student club and other professional activities and events	IT student club	-		Membership of IT student clubs, attendance at the largest event of the IT student clubs	All computer science students	30% of all computer science students at largest event	Even Fall semesters
Learns and uses technical skills not taught in class	IT 226, independent studies	IT 226	Use rubric (h)(ii)	IT 226: Project that involves a component that students are expected to learn by themselves	IT 226 students	60%	Odd Fall semesters

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

Rubric (h)(i)				
	Poor or non-existent	Developing	Developed	Exemplary
Participates in independent learning	Little or no independent learning evident in work product	Meets assignment requirements only in terms of independent learning	Shows evidence of independent learning such as gathering at least one additional external source beyond that provided or assigned, synthesizes from existing information	Collects a great deal of information-all relates to the topic; went beyond assignment requirements; applied the synthesized knowledge to real-world problems.

Rubric (h)(ii)				
	Poor or non-existent	Developing	Developed	Exemplary
Learns and uses technical skills not taught in class	Little or nothing achieved in work product pertaining to skills that were expected to be learned outside class	Meets some requirements in work product pertaining to skills that were expected to be learned outside class	Creates a successful work product based on skills that were expected to be learned outside class	Creates a successful work product that incorporates at least one aspect that was not expected, that uses the skills that were expected to be learned outside class

(i) An ability to use current techniques, skills, and tools necessary for computing practice.							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Expected level of attainment***	**Timeline**
Use a tool not taught in the classroom in the design and/or development of a moderately sized project.	IT 179, IT 226, IT 356, IT 391, Internship	Internship survey	Use rubric (i)(i)	Internship survey: Question(s) that ask of the student's ability to successfully learn and use a tool during internship that was not taught in a course	Students doing internships	80%	Data collected each semester; reviewed each fall for the previous year
Implement a software program collaboratively using version control software.	IT 226	IT 226	Use rubric (i)(ii)	IT 226: Student feedback regarding their experience with using version control software to implement group project	IT 226 students	60%	Even Spring semesters
Draws and interprets modeling diagrams.	IT 168, IT 261, IT 378,IT 391	IT 261	Use rubric (i)(iii)	IT 261: Assignment(s) that involve drawing or understanding modeling diagrams	IT 261 students	50%	Odd Spring semesters

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

Rubric (i)(i)				
	Poor or non-existent	Developing	Developed	Exemplary
Use a tool not taught in the classroom in the design and/or development of a moderately sized project	Did not complete most parts of project deliverable(s) using recommended or suitable tools (did not complete deliverable, used manual methods where tool usage was expected, etc.)	Used tools to complete only some parts of project deliverable using recommended or suitable tools (occasionally used manual methods where tool usage was expected, etc.)	Used recommended tools to complete most parts of the project deliverable while meeting most of the instructor's expectations of quality	Found tools or used recommended tools to complete most parts of the project deliverable(s) meeting or exceeding the instructor's expectations of quality

Rubric (i)(ii)				
	Poor or non-existent	Developing	Developed	Exemplary
Implement a software program collaboratively using version control software	Did not use version control software to share and coordinate program source code and documentation	Used version control only occasionally for source code and documentation, resorting to other methods for other parts	Used version control software for most program source, updated and committed to repository several times	Used version control software for most program source and documentation, all group members contributed to the repository several times

Rubric (i)(iii)				
	Poor or Non-Existent	Developing	Developed	Exemplary
Draws and interprets modeling diagrams	Unable to produce recognizable model	Can create visual model, but model does not fit problem	Creates visual model that fits problem description	Creates a well-formed and parsimonious model of problem

(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.

Performance Indicator	Delivery Methods	Courses used for Assessment	Assessment Methods	Data Needed	Assessed Groups	Expected level of attainment*	Timeline
Selects appropriate data structure and/or algorithms to solve a particular problem	IT 179, IT 279, IT 328	IT 279	Use rubric (j)(i)	IT 279: Assignment or paper that involves selecting the appropriate data structure and/or algorithm to solve a particular problem	IT 279 students	70%	Even Spring semesters
Compares and contrasts algorithms and/or data structures	IT 279, IT 328	IT 279	Use rubric (j)(ii)	IT 279: Assignment, paper or exam questions that involve comparing and contrasting algorithms and/or data structures	IT 279 students	70%	Even Spring semesters
Evaluates the space and time complexity of an algorithm or data structure using asymptotic notation	IT 179, IT 279, IT 327, IT 328	IT 279	Use rubric (j)(iii)	IT 279: Assignment, paper or exam questions that involve evaluating the space and time complexity of an algorithm using asymptotic notation	IT 279 students	70%	Even Spring semesters

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

Rubric (j)(i)				
	Poor or non-existent	Developing	Developed	Exemplary
Selects appropriate data structure and/or algorithms to solve a particular problem	Did not select an appropriate data structure and/or algorithm or did not justify selection	Selected data structure/algorithm from alternatives but did not select the most appropriate one or did not provide adequate justification for selection	Selected appropriate data structure and/or algorithm and provided reasonable justification for selection	Selected most appropriate data structure and/or algorithm and justified selection thoroughly

Rubric (j)(ii)				
	Poor or non-existent	Developing	Developed	Exemplary
Compares and contrasts algorithms and/or data structures	Provides no accurate comparison.	Identifies some correct information about similarities and differences in the algorithms, but information is incomplete and/or contains errors	Provides correct and mostly completely comparison and contrast.	Compares and contrasts data structures and/or algorithms correctly and presented conclusions about their practical usage to solve problems

Rubric (j)(iii)				
	Poor or non-existent	Developing	Developed	Exemplary
Evaluates the space and time complexity of an algorithm or data structure using asymptotic notation	Does not evaluate the space and time complexity of a given algorithm or data structure using big-O notation, evaluates based on other unreliable methods	Evaluates space and time complexity correctly using big-O notation, but does not provide tight bounds, evaluates only space or time using big-O notation but not both	Correctly evaluates a tight bound on space and time complexity of an algorithm or data structure	Proves the time and space complexity of an algorithm or data structure using big-O notation with techniques beyond expectations of the course

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

Performance Indicator	Delivery Methods	Courses used for Assessment	Assessment Methods	Data Needed	Assessed Groups	Expected level of attainment*	Timeline
Uses and writes cohesive and reusable software components	IT 179, IT 226, IT 279	IT 226	Use rubric (k)(i)	IT 226: Assignment(s) or project that involves designing and implementing software components	IT 226 students	60%	Even Spring semesters
Describes and applies common design patterns	IT 226, IT 326	IT 226	Use rubric (k)(ii)	IT 226: Homework or assignment that involves describing and/or using design patterns	IT 226 students	60%	Even Spring semesters
Identifies and describes important software engineering principles	IT 261, IT 326	Exit Exam		Exit Exam: Question(s) relating to software engineering principles	Students taking exit exam	70%	Data collected each semester; reviewed each fall for the previous year

* The expected level of attainment is measured by the minimum percentage of the assessed sample that is scored in the two maximum (Developed/Exemplary) categories of the relevant rubric.

Rubric (k)(i)				
	Poor or Non-Existent	Developing	Developed	Exemplary
Uses and writes cohesive and reusable software components	Code is not broken into components and/or code contains significant unnecessary duplication	Code is broken into components, but not with attention to reusability and avoiding code duplication	Code is broken into appropriate components that work together well and avoid duplication within the assignment, though some refactoring might improve reusability for other purposes	Components are well-designed to be reusable in a variety of relevant projects

Rubric (k)(ii)				
	Poor or Non-Existent	Developing	Developed	Exemplary
Describes and applies common design patterns	Not able to describe the specific design and use of a design pattern in general	Able to describe design pattern in general, but not identify a design pattern's application in a given problem	Able to identify design pattern in existing code, or its application to a given problem	Identifies and modifies correct design pattern to modify existing code or solve a given problem.

Rubric (k)(iii)				
	Poor or Non-Existent	Developing	Developed	Exemplary
Identifies and describes important software engineering principles	Test score <70	Test score between 70 and 80	Test score between 80 and 90	Test score above 90

2-year assessment cycle (Quick overview for implementation)

Semester	Course to be Assessed	What is assessed	Complete Assessment By
Even Fall	IT 279	a(i), a(iii), c(ii), c(iii)	Week 5 of Odd Spring semester
	Exit Exam (results from previous academic year)	c(iii), e(i), g(i), k(iii)	Week 8 of Even Fall semester
	Internship Survey (from summer just before)	i(i)	Week 8 of Even Fall semester
	IT 261	b(i), b(ii)	Week 5 of Odd Spring semester
	Student Club	h	Week 8 of Even Fall semester
Odd Spring	IT 225	a(ii)	Week 5 of Odd Fall semester
	IT 383	a(i)	Week 5 of Odd Fall semester
	IT 261	i(iii)	Week 5 of Odd Fall semester
	IT 168, IT 279 Outcome (f and j) from below just in Spring 2015	c(ii)	Week 5 of Odd Fall semester
Odd Fall	IT 226	c(i), c(iv), d(i), f(i), h(ii)	Week 5 of Even Spring semester
	Exit Exam (results from previous academic year)	c(iii), e(i), g(i), k(iii)	Week 8 of Odd Fall semester
	Internship Survey (from summer just before)	i(i)	Week 8 of Odd Fall semester
Even Spring	IT 279	f(ii), h(i), j(i), j(ii), j(iii)	Week 5 of Even Fall semester
	IT 226	i(ii), k(i), k(ii)	Week 5 of Even Fall semester
	IT 328	a(iv)	Week 5 of Even Fall semester

Review of Program Educational Objectives	
When	Procedure
Odd spring semesters	1. Assessment committee reviews and makes suggestions if any. 2. Updates are presented and discussed in faculty meeting in April of the year. 3. Approved PEOs are presented to BIAC in October meeting of the year. 4. Approved PEOs are made available to other stakeholders such as selected student groups for feedback.

Review of Student Outcomes	
When	Procedure
Odd spring semesters	1. Assessment committee reviews and makes suggestions if any. 2. Assessment committee sends report to curriculum committee and Director by end of March of the year. 3. At Director’s discretion, the updated student outcomes are tabled in faculty meeting. 4. Updated student outcomes are made available to other stakeholders such as selected student groups for feedback.