COMPUTER SCIENCE ASSESSMENT PLAN
School of Information Technology
2026

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 will have the ability to:

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Communicate effectively in a variety of professional contexts.
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.

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
(1)	▪	▪			▪
(2)	▪	▪			▪
(3)	▪		▪	▪	
(4)	▪			▪	
(5)	▪		▪	▪	
(6)	▪	▪			▪

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Target level of attainment***	**Timeline**
a. Conducts system analysis using common modeling techniques to assess a problem.	IT 261, IT 326, IT 378,	IT 261	Use rubric 1(a)	IT 261: Assignment(s) that deal with using models to analyze a problem	IT 261 students	50%	Even Fall semesters
b. Gathers requirements for a given problem	IT 261, IT 378	IT 261	Use rubric 1(b)	IT 261: Assignment(s) that deal with gathering requirements for a given problem	IT 261 students	50%	Even Fall semesters
c. Uses number systems and Boolean logic to describe how computing systems work	IT 225, IT 279	IT 225	Use rubric 1(c)	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

* The target 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 1(a)				
	Poor or Non-Existent	Developing	Developed	Exemplary
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 reasonably fits problem description	Creates a well-formatted and efficient visual model that represents a good fit for the problem

Rubric 1(b)				
	Poor or Non-Existent	Developing	Developed	Exemplary
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

Rubric 1(c)				
	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 represent 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

2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Target level of attainment***	**Timeline**
a. Conducts system design using common modeling techniques to solve a problem.	IT 261, IT 326, IT 378,	IT 261	Use rubric 2(a)	IT 261: Assignment(s) that deal with using models to design a solution to a given problem	IT 261 students	60%	Even Fall semesters
b. Writes a computer program that solves a problem	IT 168, IT 179, IT 279, IT 328, IT 340, IT 353, IT 356, IT 383	IT 168, IT 279	Use rubric 2(b)	IT 168 and IT 279: Programming assignment(s)	IT 168 students IT 279 students	50% (IT 168) 70% (IT 279)	IT 168: Even Fall semesters IT 279: Even fall semesters
c. Evaluates alternative solutions for a given problem	IT 279	IT 279	Use rubric 2(c)	IT 279: Homework or paper that deals with proposing or evaluating multiple solutions to the same problem	IT 279 students	70% (IT 279)	IT 279: Even fall semesters
d. Writes test cases for a problem and/or implementation	IT 168, IT 326	IT 326	Use rubric 2(d)	IT 326: Assignment(s) that ask students to write test cases for a given problem or implementation	IT 326 students	60%	Odd Fall semesters
e. Measures computing performance	IT 179, IT 225, IT 279, IT 328, IT 356, IT 378, IT 383	IT 279, IT 383	IT 279: Use rubric 2(e-1) IT 383: Use rubric 2(e-2)	IT 279: Homework or paper that involves efficiency analysis IT 383: Assignment(s) or exam questions 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

Rubric 2(a)				
	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 represent 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 2(b)				
	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, including unseen data, and does so efficiently with well-designed code.

Rubric 2(c)				
	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 2(d)				
	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

Rubric 2(e-1)				
	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.

Rubric 2(e-2)				
	Poor or Non-Existent	Developing	Developed	Exemplary
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

3. Communicate effectively in a variety of professional contexts.							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Target level of attainment***	**Timeline**
a. Communicates effectively in a variety of professional contexts orally	IT 191, IT 326, IT 378, COM 110, IT 398	IT 326	Use rubric 3(a)	IT 326: Oral Presentation	IT 326 students	60%	Odd Fall semesters
b. Communicates effectively in a variety of professional contexts in writing	IT 191, IT 279, IT 326, IT 327, IT 328, IT 378, IT 398, ENG 101, ENG 249	IT 279	Use rubric 3(b)	IT 279: Written paper	IT 279 students	70%	Even Spring semesters

* The target 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 3(a)				
	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 3(b)				
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 aimed at the audience, occasionally too detailed or too vague	Mostly aimed 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	Appropriate number of well-chosen visual aids that enhance communication

4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Target level of attainment***	**Timeline**
(a) Identify laws that affect Information Technology	IT 214	IT 214	Use rubric 4(a)	Exam: Question(s) relevant to identifying whether existing software programs can be used in a specific setting based on their licenses	IT 214 students	60%	Data collected odd Fall
(b) Identifies sections of a professional code of ethics that apply to a given situation	IT 214	IT 214	Use rubric 4(b)	Exam: Question(s) that relate sections of a professional code of ethics to a given situation	IT 214 students	60%	Data collected odd Fall
(c) Identify prevailing ethical principles	IT 214	IT 214	Use rubric 4(c)	Exam: Question(s) relevant to various prevailing ethical principles	IT 214 students	60%	Data collected odd Fall

* - The target 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 4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.				
	Poor or Non-Existent	Developing	Developed	Exemplary
(a) Identify laws that affect the IT industry and profession	Can't identify any laws that affect the IT industry	Identifies a few laws that affect the IT industry	Identifies laws that affect the IT industry	Identifies laws that affect the industry and can identify laws that will affect a particular system
(b) Identify elements from a professional code of ethics	Can't identify any elements from a professional code of ethics	Identify some the elements from a professional code of ethics	Identify most of elements from a professional code of ethics	Identify all elements from a code of ethics
(c) Identify prevailing ethical principles	Cannot identify any prevailing ethical principles	Identifies some prevailing ethical principles	Identifies most prevailing ethical principles	Identifies all prevailing ethical principles

5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Target level of attainment***	**Timeline**
(a)Actively participates on the team	IT 261, IT 326, IT 378, IT 391, IT 191	IT 326	Use rubric 5(a), 5(b)	IT 326: Peer and group reviews from group assignment(s) or projects	IT 326 students	60%	Odd Fall semesters
(b)Completes team assignments on time							

* The target 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 5(a)(b)				
	Poor or Non-Existent	Developing	Developed	Exemplary
(a) Actively Participates in team activities	Does not contribute to discussions, does not let others express opinions	Contributes occasionally to team activities	Contributes equally in team activities	Contributes a higher share to team activities without taking over the team
(b) Completes team assignments on time	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

6. Apply computer science theory and software development fundamentals to produce computing-based solutions							
Performance Indicator	**Delivery Methods**	**Courses used for Assessment**	**Assessment Methods**	**Data Needed**	**Assessed Groups**	**Target level of attainment***	**Timeline**
a. Uses and writes cohesive and reusable software components	IT 179, IT 326, IT 279	IT 326	Use rubric 6(a)	IT 326: Project that involves designing and implementing software components	IT 326 students	60%	Even Spring semesters
b. Compares and contrasts data structures and algorithms, selecting the appropriate ones to solve a particular problem	IT 179, IT 279, IT 328	IT 279	Use rubric 6(b)	IT 279: Assignment, paper, or exam questions that involve selecting the appropriate data structure and/or algorithm to solve a particular problem	IT 279 students	70%	Even Spring semesters
c. 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 6(c)	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
d. Describes how concepts of computability are related to computing solutions	IT 328	IT 328	Use rubric 6(d)	IT 328: Assignment(s) relating concepts of computability to practical computing	IT 328 students	70%	IT 328: Even Spring

* The target 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 6(a)				
	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 6(b)				
	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 6(c)				
	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

Rubric 6(d)				
	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 problems, and the complexity and computability implications of the nondeterministic model under different constraints
Understand Regular Languages	Does not know what they are	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 they are	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 they are	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

	CS-2-year assessment cycle (Quick overview for implementation)		
Semester	Course to be Assessed	What is assessed	Target level of attainment
Even Fall	IT 261	1(a), 1(b), 2(a)	50%, 50%, 60%
	IT 279	2(b), 2(c), 2(e)	50%,70%,70%
	IT168	2(b)	50%
Odd Spring	IT 225	1(c)	50%
	IT 383	2(e)	70%
Odd Fall	IT 326	2(d), 3(a), 5(a), 5(b)	60%
	IT 214	4(a), 4(b), 4(c)	60%
Even Spring	IT 326	6(a)	60%
	IT 279	3(b), 6(b), 6(c)	70%,
	IT 328	6(d)	70%

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.