B.S. Computer Science Three-Year Assessment Plan for Student Outcomes

Year	Outcomes Assessed (Abbreviated Form)	Courses	Data Collected	Continuous Improvement
Year 1 (2015-2016)	(a) Application of fundamental knowledge	CSC 130, 133, 134, 135, 137, 138, and 139	Direct assessment in course-embedded exam questions, assignments, and projects Supervisor evaluation of student interns	Analyze results of assessment of SOs (a)-(d) and make recommendations for the performance indicators that are below the standard (target success rate of 70%). Implement previous year's faculty recommendations for performance indicators for SOs (g) and (h) that are below minimum and re-assess these indicators.
	(b) Computer system development cycle	CSC 131, 137, 138, 139, and 190/191		
	(c) Application of software development principles	CSC 131, 133, 138, and 190/191		
	(d) Application of skills, techniques, and tools for computing practice	CSC 133, 134, 135, 137, 139, and 195/195A		
Year 2 (2016-2017)	(e) Team work	CSC 131, 190/ 191, and 195/195A	Instructor evaluation Student self-assessment and reflection Supervisor evaluation of student interns	Analyze results of assessment of SO (e) and SO (f) and make recommendations for performance indicators below standard. Implement previous year's faculty recommendations for performance indicators for SOs (a) - (d) that are below minimum and, re-assess these indicators.
	(f) Oral Communication	CSC 131, 190/191, and 195/195A	Faculty evaluation of student oral presentations using a rubric Supervisor evaluation of student interns	
Year 3 (2017-2018)	(g) Professional, ethical, and security issues and responsibilities	CSC 138, 190/191, and 195/195A; PHIL 103	Course-embedded exam questions Student surveys Faculty evaluation of written essays Supervisor evaluation of student interns	Analyze results of assessment of SO (g) and SO (h) and make recommendations for performance indicators below standard. Implement previous year's faculty recommendations for performance indicators for SO (e) and SO (f) that are below minimum and. re-assess these indicators.
	(h) Written communication	CSC 190/191 and 195/195A	Faculty evaluation of written reports using a rubric Supervisor evaluation of student interns	

The expected level of attainment for each of the student outcomes.
For each performance indicator, the percentage of student responses meeting or exceeding the performance standard is computed. Then, for each outcome, the average of the percentages for all relevant performance indicators is computed. If the average percentage for an outcome is greater than or equal to 70%, the outcome is considered to be satisfied. Although, in the past, the minimum standard was set at 75%, the faculty decided in 2013-2014 to use a 70% standard since it is common practice to consider a score of 70% to be a passing grade.

Correspondence between Upper Division Required Courses and Student Outcomes

Outcomes / Courses	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
CSC 130	X			X				
CSC 131	X	X	X	X	X	X	X	X
CSC 133	X	X	X	X				
CSC 134	X			X				
CSC 135	X	X	X	X				
CSC 137	X	X		X				
CSC 138	X	X	X	X		X		
CSC 139	X	X		X		X		
CSC 190/191	X	X	X	X	X	X	X	**X**
CSC 192 & CSC 194						X	X	
CSC 195 & CSC 195A	X	X	X	X	X	X	X	X
CSC 198 & CSC 199	X	X		X		X		

Student Outcomes and Performance Indicators

	Performance Indicator	Core Course
(a) Apply fundamental knowledge of mathematics, algorithmic principles, computer theory, and principles of computing systems in the modeling and design of computer-based systems that demonstrate an understanding of tradeoffs involved in design choices.	a-1. Understand fundamental algorithms and essential data structures.	CSC 130
	a-2. Understand trade-offs in the selection of algorithms and data structures.	CSC 130
	a-3. Understand and apply mathematical transformations and algorithms for 2D graphics.	CSC 133
	a-4. Understand and use relational databases.	CSC 134
	a-5. Understand distinctive features of the design of programming languages.	CSC 135
	a-6. Demonstrate knowledge of abstract machines, languages, and grammars.	CSC 135
	a-7. Understand and apply the logic programming paradigm.	CSC 135
	a-8. Understand and apply the functional programming paradigm.	CSC 135
	a-9. Demonstrate the ability to calculate performance parameters, such as, circuit propagation delay, memory latency, speedup, etc.	CSC 137
	a-10. Understand network architecture, layered model, and protocol stacks.	CSC 138
	a-11. Demonstrate the working knowledge of network management including monitoring, measurement, analysis, and control.	CSC 138
	a-12. Understand principles of concurrency and tradeoffs in synchronization approaches, analysis, and control.	CSC 139
	a-13. Understand deadlocks and their solutions.	CSC 139
	a-14. Understand principles of resource management.	CSC 139

(b) Analyze a problem, specify the requirements, design, implement, and evaluate a computer-based system, process, component, or program that satisfies the requirements.	b-1. Understand and apply modeling and analysis techniques.	CSC 131, 190/191
	b-2. Understand and apply requirements engineering process.	CSC 131, 190/191
	b-3. Understand and apply design principles.	CSC 131*, 190/191
	b-4. Understand and apply proper testing techniques	CSC 131*, 190/191
	b-5. Understand and apply project management processes and tools.	CSC 131, 190/191
	b-6. Demonstrate the ability to design and analyze basic and complex hardware components.	CSC 137
	b-7. Understand and apply error detection and correction, flow control, and congestion control principles.	CSC 138
	b-8. Understand and apply synchronization mechanisms to the critical section problem and to the process coordination.	CSC 139
(c) Apply design and development principles in the construction of software systems of varying complexity.	c-1. Understand and use software metrics.	CSC 131
	c-2. Understand and use object-oriented design.	CSC 131*, 133
	c-3. Understand and use design patterns.	CSC 133
	c-4. Understand and use verification and validation techniques.	CSC 131, 190/191
	c-5. Understand and apply documentation standards.	CSC 131, 190/191
	c-6. Understand and apply semi-formal modeling languages, such as, UML, in requirement specification and design.	CSC 190/191
	c-7. Demonstrate the ability to develop communication protocols and networking applications.	CSC 138

(d) Use current skills, techniques, and tools necessary for computing practice.	d-1. Implement event-driven GUI applications.	CSC 133
	d-2. Demonstrate competence in using SQL.	CSC 134
	d-3. Demonstrate competence in programming in a variety of programming paradigms.	CSC 135
	d-4. Demonstrate competence in language scanning and parsing.	CSC 135
	d-5. Demonstrate the ability to use hardware design simulation tools.	CSC 137
	d-6. Demonstrate competence in system programming in Unix/Linux environments.	CSC 139
(e) Function effectively as a team to accomplish a common goal.	e-1 Cooperate and collaborate as a team member.	CSC 191
	e-2. Communicate and listen; keep teammates informed.	CSC 191
	e-3. Face conflicts and resolve most differences.	CSC 191
	e-4 Contribute equally as a participant in the project.	CSC 191
(f) Understand professional, ethical, and security issues and responsibilities.	f-1. Know, understand, and practice professional codes of conduct (*i.e., ACM Code of Ethics and Professional Conduct, IEEE Code of Ethics, ACM/IEEE Software Engineering Code of Ethics and Professional Practice.)	PHIL 103, CSC 190/191
	f-2 Understand need for and use of proper security features.	CSC 138
	f-3. Be able to evaluate the ethical dimensions of a computer solution to a problem.	PHIL 103
	f-4. Understand moral and ethical dimensions of a computer solution to a problem.	PHIL 103

(g) Write effectively.	g-1. Focus – responds to the questions asked.	CSC 191
	g-2. Structure – well-organized, consistent style, and smooth transitions	CSC 191
	g-3 Sentence Structure – use of language: clearly communicates ideas, uses correct syntax, grammar, and spelling. Word Choice – use and placement of words are appropriate.	CSC 191
	g-4. Paragraph Structure – well-written paragraphs on topic and understandable.	CSC 191
	g-5. Problem Statement – objective, nature of challenges, and value of project are clear; purpose is clear.	CSC 191
	g-6. Design Requirements – specifications complete and design constraints identified	CSC 191
(h) Give effective oral presentations.	h-1. Effective style and delivery.	CSC 131, 191
	h-2. Correct language and vocabulary	CSC 131, 191
	h-3. Good organization	CSC 131, 191
	h-4. Clear communication of technical content	CSC 131, 191
	h-5. Project related issues	CSC 191