

Student Learning Outcomes for Computer Science

Upon completion of the degree, Computer Science graduates will:

Outcome 1: Understand classical algorithmic processes and data structures and be able to perform simple analysis of algorithms—embedded tests, MFAT sub-area Discrete Structures and Algorithms

Outcome 2: Be proficient in one high-level programming language and have basic skill levels in a variety of programming languages—embedded tests, FAT sub-area Programming Fundamentals

Outcome 3: Understand the basics of logic design and computer organization and be aware of multiple architecture approaches and numerical limitations—embedded tests, MFAT sub-area Systems (Architecture, Operating Systems, Networking, Database)

Outcome 4: Be competent in techniques of discrete mathematics and understand the theoretical foundations of computing—embedded tests, MFAT sub-area Discrete Structures and Algorithms

Outcome 5: Understand the steps of the software development process and the activities/products appropriate to each—PROGRAM RUBRIC

Outcome 6: Know the major issues in the design and implementation of major computing artifacts such as operating systems, programming languages, graphical user interfaces or systems programming, and databases, networks, or compilers—embedded tests, MFAT sub-area Systems (Architecture, Operating Systems, Networking, Database)

Outcome 7: Be able to adapt to changing development platforms and design/implementation tools—PROGRAM RUBRIC

Outcome 8: Communicate effectively on technical matters in both oral and written forms, work well within a team, and understand the social/ethical issues of computing—PROGRAM RUBRIC

Curriculum Matrix

I = Introduced, D = Developed & Practiced with Feedback, M = Demonstrated at the Mastery Level appropriate for graduation in this program

Course	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6	Outcome 7	Outcome 8
CS 141 – Discrete Mathematics for Computer Science 1	I		I	I				I
CS 150 - Introduction to Computer Science	I	I			I		I	
CS 151 - Introduction to Software Development	D	D			D			
CS 241 – Discrete mathematics for CS II				D				
CS 266 - Computer			D				I	

Organization and Assembly Language								
CS 321 -Data Structures	M	M						D
CS 340 – GUI						D	D	
CS 350 – Systems Programming		D				D	D	
CS 407 – Introduction to Numerical Analysis I		D	M				D	
CS 410 - Elements of Computer Architecture			M				D	D
CS 420 -File Management						M	D	
CS 421 – Database Management System Design						M	M	
CS 431 – Computer Networks / Data Communication						M	M	
CS 450 - Organization of Programming Languages		D				M	M	
CS 451 – Compiler Theory		D				M	M	
CS 460 - Software Engineering I		M			M		M	M
CS 461 - Software Engineering II		M			M		M	M
CS 470 - Theory of Computing				M				
CS 495 -CS Professional Seminar								M

Rubric for PLOs 5, 7 & 8

	Teamwork	Design and Planning	Responsiveness to Client Needs and Concerns (eval made by client)	Effectiveness and Workability of Final Product
Far Exceeds Expectations 4	All members of the team perform at the highest level and contribute throughout the year-long process. Team needs very little prompting to undertake work; team demonstrates	The team shows remarkable skill in crafting a clear outline of the program and the steps needed to undertake the development of the software. Team shows proactive thinking in anticipating possible setbacks or may	Team is constantly engaged with the client in all aspects of the design, planning, and execution of the program. Team responds quickly to concerns or proactively reaches out as opportunities for	The final product exceeds the expectations of the client. Product is extremely easy to use and well crafted. Product is seen as having a marked

	initiative on their own. Highly engaged work team.	proactively identify hidden opportunities for further development. Team is highly adaptive in either situation.	refinement or working issues arise. Team can anticipate needs on behalf of the client.	impact in the community.
Good 3	All members of the team work well together. Team takes cues from the teacher and undertakes tasks as needed. All are engaged and responsive to each other.	The team is able to develop a basic plan and sticks with it throughout the year. Problems are handled with care and continued focus on the timeline.	Team engages with the client as issues or questions arise. Team uses feedback to guide product development.	Final product meets the client's expectations but does not offer substantive impact. Product is useful and reliable.
Minimal Competence 2	Some members may try their best but it is clear not everyone is putting in the same effort. Lack of communication between members makes for occasional misunderstandings or loss of time/effort. Team constantly seeks help from the instructor as opposed to achieving operational independence.	Team comes up with a basic timeline and may not have fully anticipated any problems that tend to disrupt working projects. Engagement with the project may not be consistent over time but the product is delivered by deadline.	Team may not always check with the client on needs. Team may make assumptions on the part of the client; team may not always consider client feedback.	Final product may work under certain conditions but may not be efficient or error free. Program may be hard to use for the average lay person.
Unable to meet standards 1	Team cannot function at all as a group. Need constant intervention and directives by the teacher.	The team lacks a plan for development. Work is not consistent throughout the year. It appears last ditch efforts were made towards the end to meet the deadline; or the product may be delivered late.	Product plan and design shows no thought about what is needed by the client. Team does not communicate well with the client.	Product does not fully function or address the needs of the client.