

Course Outcome Guide (COG)

Approved 13 September 2012

Course:	CSCI 160 – Computer Science I	Credits:	3	Instructor:	TBD
Course Description:	An introduction to computer science including problem solving, algorithm development and structured programming in a high-level language. Emphasis on design, coding, testing and documentation of programs using accepted standards of style.				
Concepts and Issues	Process Skills	Assessment Tasks	Intended Outcomes		
			Course	General Education or Program	Institutional
<p>Basic concepts of programming language</p> <p>Functions</p> <p>Arrays</p> <p>STL strings</p> <p>Structures</p> <p>Compilers</p> <p>Pseudocode</p> <p>Debugging tools</p>	<p>Follow the software development process (requirements analysis, design, implementation, and test) in the development of small programs.</p> <p>Use an understanding of cultural differences in user populations and global software design requirements in order to design effective software.</p> <p>Employ good software engineering practices and good software design, always applying Software Engineering Code of Ethics as determined by Association for Computing Machinery (ACM).</p> <p>Construct appropriate user interfaces for simple programs, and design systems with minimal complexity and maximal functionality.</p> <p>Analyze and construct efficient and effective</p>	<p>*Participate in class discussions and activities demonstrating knowledge of subject matter.</p> <p>*Complete examinations demonstrating acceptable skill level of concept and process.</p> <p>*Complete textbook readings, questions and problems (both individually and collaboratively) demonstrating acceptable skill levels of concept and process.</p> <p>* Design, construct and test your final project.</p>	<p>Students demonstrate an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs including exposure to simple sorting, searching, and searching.</p> <p>Students demonstrate an ability to use current techniques, skills, and tools necessary for computing practice</p> <p>1. Final Problem: student on paper walks through a C++ code sample by hand that includes loops, functions, selection, parameters, arrays strings etc and correctly determining the result.</p> <p>2. Final Project: game of life programming problem in C++ or similar.</p> <p>3. Final Problem:</p>	<p>1.Mathematics-including numeration literacy and the knowledge and use of statistical and logical processes.</p> <p>2.Analytical-gathering, organizing, and evaluating information</p> <p>3.Analogical-using former knowledge to help comprehend and explain new situations</p> <p>4.Critical Thinking-the ability to identify ad define criteria, understand biases, and construct objective judgments.</p> <p>5.Problem solving-the ability to analyze situations and synthesize solutions.</p>	<p>1. Students will demonstrate effective communication skills.</p> <p>2. Students will use reasoning skills to analyze and solve problems.</p>

	<p>algorithms and translate to appropriate control structures in an implementation language. Effectively use software development tools including libraries, compilers, editors, linkers and debuggers.</p>		<p>student can write a function that generates a specified result such as a function that accepts a string and returns its reverse. 4. Project: statistics project that includes bubble sort, mean, median, standard deviation, mode Students demonstrates an ability to apply design and development principles in the construction of software systems of varying complexity through the use of structure charts, good names, and functions that implement a single task.</p>		
--	---	--	---	--	--