Assessment: Course Four Column

Fall 2018

El Camino: Course SLOs (MATH) - Computer Sciences

ECC: CSCI 1:Problem Solving and Program Design Using C++

Course SLOs	*Assessment Method Description*	*Results*	*Actions*
SLO #1 Writing Algorithms - Students will write correct and detailed algorithms. (Properly analyze a problem using top down design, and write an algorithm that can be translated into computer code.) **Course SLO Status:** Active **Course SLO Assessment Cycle:** 2014-15 (Fall 2014), 2018-19 (Fall 2018) **Input Date:** 11/19/2013	**Exam/Test/Quiz -** Exam/project given during the semester. The topics included writing algorithms for arrays, functions etc. Grading Rubric: 10 points - Concise description of reasonable problem solving technique exists and the progress made from the process is clear from the description. 8 points - description of reasonable problem solving technique exists and the progress made from the process is clear from the description, but could be simplified. 5 points - description of problem solving technique exists, but misses' key details. 3 points - description exists, but at a very basic level 0 points - no description exists Grading was done out of 10 points. **Standard and Target for Success:** Score of 80% **Related Documents:**	**Semester and Year Assessment Conducted:** 2018-19 (Fall 2018) **Standard Met? :** Standard Met Table below gives score distributions. Score range — Number of students in that range — Percentage of Students in each range (%) 90 % to <=100% — 56 — 42.1 80% to <=90% — 13 — 9.8 70% to <=80% — 19 — 14.3 60% to <=70% — 10 — 7.5 50% to <= 60% — 9 — 6.8 <=50% — 26 — 19.5 Total — 100.0 The table below shows the statistics of above data: Statistical Property — Value of Statistical Property Highest — 100% Average — 74%	**Action:** CS faculty adds an Algorithm writing components to all CS1 Labs. CS department should come up with a procedure to audit, whether this is being done in all CS1 classes or not. This may need helping some adjunct faculty to learn procedures to grade such written algorithms, but that is worth doing. (12/18/2022) **Action Category:** Teaching Strategies **Action:** Add one or more writing assignment in CS1 in first week of classes. Item 2 b above describes that. Prompt feedback is to be given to weaker students and corrective strategy for them is to be designed. (03/06/2020) **Action Category:** Teaching Strategies

Course SLOs	Assessment Method Description	Results	Actions
	Fall 2014 SLO Report for CSCI 1.docx	Median 90% Standard Deviation 33% Lowest 0% Data Analysis and Interpretation of results First table shows that about 73.7 % students, scored 70% or higher in the SLO #1 test. Thus, computer science department standard, that 70% or more students score 70% or higher in an SLO test was met. However, computer science department should not ignore the fact that about 20% (1/5th) student population really failed SLO test, and student data distribution is tri-modal. There are about 40% of the student in range 90 to 100 % scoring range. Then there are 32% (1/3rd) students in range 70 to 90%. If we consider D (<70%) as unsuccessful completion, then failure rate is even higher (26%). At the same time, however, one has to be careful in considering the result from one assessment (like one SLO question) as a global reflection of student learning in CS1. One result is clear. This SLO was about writing algorithms. Doing that requires reasonable writing skills. There is about 1/4th to 1/5th of student population, whose writing skill is poor quality. But it is also encouraging that 4/5th to 3/4th of students have average to excellent writing skills. Computer Science department should consider three pronged actions to improve algorithm writing skills of 1/4th of the student population which did poorly in this SLO. These three-pronged actions are divided in three categories below: 1. Long term Macro actions: a. Working with El Camino advisors CS faculty should encourage weaker students (poor writing and algorithmic skills) to take CS7 first, which has a specially designed section on algorithms. b. Perhaps funds may be arranged to have orientation for students who are registered for CS1. In this orientation, students can be told about entry path to computer science (through CS7 – Gentle Entry, through CS1 – Labor Intensive Entry). This will help many students chose	

Course SLOs	Assessment Method Description	Results	Actions
		a proper entry path to computer science. One loophole in this approach is that non-majors take CS1 because their major requires it. In CS1 orientation, CS department can emphasize to these non-majors, that the reason their major requires CS1 is because in their professional life, at some juncture, it is anticipated that they may need to do software development or maintain an existing software. Thus, if success in CS1 (even as a non-major) would affect their career, then entry through CS7 is a very good way to do it. c. CS department should start offering CS7 in local high schools. That way the gentle entry to CS is finished in high schools and entry to CS1 would become easier. d. CS7 should be offered in summer and winter semesters, so that target population can take it in a shorter span of time. That CS7 background would help students when they take regular semester CS1 course. 2. Micro actions that can be implemented in CS1, and in classes that students may take before CS1: a. Unfortunately, we have no control over it, but we can talk to our faculty colleagues in Math, whether they would consider giving students credit for writing down procedure to solve a Math problem in Math HomeWorks. This would strengthen student's writing skills. b. In first week in CS1, students do a writing assignment. This could be writing simple algorithms, writing answers to questions, or writing a short CS essay. c. Writing algorithm must be done in each CS1 lab and there should be enough points assigned to it, to inspire students. d. Computer Science TA's are specially trained to help students solve problems in writing algorithms. e. El Camino ACM offers specially designed workshops in writing algorithms (No coding. Just writing algorithms. Flow charts can be used as well). College level efforts are required if student lack of success is due to one or more of following factors: 1. Lack of engagement. 2. Demanding work and college schedule. 3. Borderline success in pre-requisite class or having done	

Course SLOs	Assessment Method Description	Results	Actions
		such class so long ago that due to lack of use the pre-requisite material has been forgotten. 4. Sudden change in student's life condition that required attention and time resources to be redirected from studies towards resolution of such condition. We have no way of knowing as how many students are affected by which factor(s) above. (03/06/2019) **% of Success for this SLO:** 100 **Faculty Assessment Leader:** Satish Singhal **Faculty Contributing to Assessment:** Satish Singhal, Prof. Victor Matos, Prof. Massoud Ghaym, Prof. Abbas Dehkhoda, Prof. Hathairat Rattanasook	
		Semester and Year Assessment Conducted: 2014-15 (Fall 2014) **Standard Met? :** Standard Not Met Total students assessed 130 86 students or 66% scored 80% to 100%, 32 students or 25% scored in the range of 60% to 79% and 12 students or 9% scored below 60% on the assessment. Interpretation of results For the students who met the target, I think they communicated well with the instructor, understood class lectures, studied the supporting materials and learned overall art of developing algorithms. About 1/3rd of class did not meet SLO standard of 80 % that was set. That could have been due to combination of factors. Typical factors we have seen hindering student success in community colleges and Computer Science are: 1. Lack of engagement, due to factor such as Computer Science not being student's major. 2. Demanding work and college schedule. 3. Borderline success in pre-requisite class or having done such class so long ago that due to lack of use the pre-requisite material has been forgotten. 4. Sudden change in student's life condition that required attention and time resources to be redirected from studies towards resolution of such condition. (02/26/2015)	**Action:** Action Plan The most important thing we have realized is that 80% or a B- grade set as a success standard is too high for a class such as CSCI 1 because CSCI 1 comprises students from multiple disciplines. Their engagement in class cannot be of same level of the Computer Science students. Thus in future we would lower the success standard to 70%. If distribution in the range 60 to 80% is linear then the success rate rises to about 79% which would be reasonable for the class such as CSCI 1, which is first Computer Science discipline class. (09/26/2018) **Action Category:** SLO/PLO Assessment Process **Follow-Up:** We have lowered the success standard in CS1 to 70% for subsequent SLO assessments (03/11/2019)

Course SLOs	Assessment Method Description	Results	Actions
		Faculty Assessment Leader: Satish Singhal **Faculty Contributing to Assessment:** Satish Singhal, Massoud Ghyam, J L:eon, Sophia Sherif, Dave Akins	
	Project - Temporary or hourly workers get paid by hours worked. Labor department laws regarding temporary help are below: If hours worked are up to 40 hours or less then workers salary is hours worked multiplied by hourly pay rate. However, if hours worked exceeds 40 then hours above 40 must be paid by the rate of 1.5 times of regular hourly rate. In addition, employers can withhold social security tax at federally mandated rates. Write following user defined functions. 1. double getGrossSalary (double hoursWorked, double PayRate); This function is called from the main function that already has user data for hours worked and pay rate, inputted by the user. Function applies the algorithm below and returns to the calling block the gross salary. 2. Write a function with below header: double getNetSalary(double grossSal, double TaxRate); Function uses the gross salary determined by the function #1 and a user entered tax rate. Function deducts the tax from the gross salary and computes the net salary and returns it to the calling block. 3. Write a main function for integration of above functions.		

Course SLOs	Assessment Method Description	Results	Actions
	Grading rubric: Analyzing Software 2 points Writing Algorithm/Pseudo code 2 points Writing source code 2 points Fixing and documenting bugs 2 points Testing and documenting program 2 points Total 10 points **Standard and Target for Success:** 70% of the student score 70% or higher in SLO test.		

ECC: CSCI 14:Computer Programming in Python for Computer Science

Course SLOs	*Assessment Method Description*	*Results*	*Actions*
SLO #1 Writing Algorithms - Student will write and correct detailed algorithms, some of them would include text processing. (Properly analyze a software problem using top down design, and write related algorithm that can be translated into computer program in Python). **Course SLO Status:** Active **Course SLO Assessment Cycle:** 2018-19 (Fall 2018) **Input Date:** 08/01/2017	**Exam/Test/Quiz -** A question was given on the final test to evaluate student's knowledge in designing algorithms. **Standard and Target for Success:** Scoring minimum of 8 out of 10 is considered full understanding, and score of 6 or 7 was considered most understanding. It is expected that 80% of the students will pass successfully. **Related Documents:** CSci14_SLO_1_Fall18.docx	**Semester and Year Assessment Conducted:** 2018-19 (Fall 2018) **Standard Met? :** Standard Met Students performed well and all students scored above 7/10 of which 10 people scored 9 or 10 and 3 people scored 7. (03/08/2019) **% of Success for this SLO:** 90 **Faculty Assessment Leader:** Massoud Ghyam **Faculty Contributing to Assessment:** Massoud Ghyam	**Action:** I plan to require more algorithm development assignments without programming at the beginning of the semester to enforce the importance of designing solutions first prior to coding. (03/08/2019) **Action Category:** Teaching Strategies

ECC: CSCI 16:Assembly Language

Course SLOs	*Assessment Method Description*	*Results*	*Actions*
SLO #1 Developing PC Assembly Language Code - Students will design, code, compile, test and document programming solutions to problems by developing PC assembly language code that makes direct use of processor instructions, interrupts, registers, the stack, as well as existing macro and procedure libraries. **Course SLO Status:** Active **Course SLO Assessment Cycle:** 2014-15 (Fall 2014), 2018-19 (Fall 2018) **Input Date:** 11/19/2013	**Laboratory Project/Report -** Students completed multiple (more than three) programming projects, working in the lab and at home on their own computers. **Standard and Target for Success:** 80% of students will be able to complete and be able to explain the code that they have written. **Related Documents:** CSCI 1 0134 2014 Fall Pgm01.pdf CSCI 1 0134 2014 Fall Pgm02.pdf CSCI 1 0134 2014 Fall Pgm03.pdf	**Semester and Year Assessment Conducted:** 2014-15 (Fall 2014) **Standard Met? :** Standard Met At the end of the term, there were only 12 students in the class. Two of the students, for whatever reason, did not do the work. Hence, the real data is only for 10 students. Within this group, all 10 demonstrate good knowledge of 80% or more of the material. (02/05/2015) **Faculty Assessment Leader:** Ralph Taylor	**Action:** The next time I teach this course, I will have the material set for the entire semester. This semester, it had been 9 years since the course was last offered and the current operating systems that we are using would not allow the use of some of the basic components of assembly language. We need to develop some way of getting around this problem. Also, I will be giving at least 10, and up to 20, quizzes though-out the semester. (09/07/2015) **Action Category:** Teaching Strategies **Follow-Up:** This has been resolved using Visual Studio's MASM component. It is the best tool we have and will support the majority of the basic components of assembly language. (03/14/2018)
	Laboratory Project/Report - The following problem was given to students in an assignment: Create a procedure named CalcGrade that receives an integer value between 0 and 100, and returns a single capital letter in the AL register. Preserve all other register values between calls to the procedure. The letter returned by the procedure should be according to the following ranges:	**Semester and Year Assessment Conducted:** 2018-19 (Fall 2018) **Standard Met? :** Standard Met Of 22 students assessed, 18 scored 70% or above on the problem. That is a 82% success rate. This problem was part of the 4th of 8 assignments in the course and was due during the eleventh week of the 16 week term. (02/28/2019) **% of Success for this SLO:** 82 **Faculty Assessment Leader:** Edwin Ambrosio	**Action:** Will review the effectiveness of this assessment tool/method and the relevance of the SLO (02/28/2020) **Action Category:** SLO/PLO Assessment Process

Course SLOs	Assessment Method Description	Results	Actions
	Letter Grade Score Range A 90 to 100 B 80 to 89 C 70 to 79 D 60 to 69 F 0 to 59 Write a test program that generates 10 random integers between 0 and 100, inclusive. Each time an integer is generated, pass it to the CalcGrade procedure. Use the book's library and display each integer and its corresponding letter grade. **Standard and Target for Success:** It is expected that 80% will score 70% or above on this problem. 0 –No understanding The student is unable to design, code, compile, test and document programming solutions to problems by developing PC assembly language code that makes direct use of processor instructions, interrupts, registers, the stack, as well as existing macro and procedure libraries. 1 –Some understanding The student is somewhat able to design, code, compile, test and document programming solutions to problems by developing PC assembly language code that makes direct use of processor instructions, interrupts, registers, the stack, as well as existing macro and procedure libraries.		

Course SLOs	Assessment Method Description	Results	Actions
	2 –Most understanding The student is generally able to design, code, compile, test and document programming solutions to problems by developing PC assembly language code that makes direct use of processor instructions, interrupts, registers, the stack, as well as existing macro and procedure libraries. 3- Complete understanding The student is able to completely design, code, compile, test and document programming solutions to problems by developing PC assembly language code that makes direct use of processor instructions, interrupts, registers, the stack, as well as existing macro and procedure libraries.		

ECC: CSCI 2:Introduction to Data Structures

Course SLOs	*Assessment Method Description*	*Results*	*Actions*
SLO #1 Programming Solutions - Students will design, code, compile, test and document a programming solution to a problem involving the basic data structures: lists, stacks, queues, trees, and related abstract data types. **Course SLO Status:** Active **Course SLO Assessment Cycle:** 2014-15 (Fall 2014), 2018-19 (Fall 2018) **Input Date:** 11/19/2013	**Project -** The purpose of assessment is to use a stack ,queue and binary search tree class designed to store data in respective data structures and then solve either a palindrome problem or sort and process data in required form. Grading Rubric Designing a solution. This includes you submitting a design document that would include, input, output, and analysis that what algorithms, strategies, class designs would be necessary for software to create an output from given input. 2 Points Coding the above design. 2 Points Compiling the above design and removal of compile, logic, and runtime errors. 2 Points Testing the solution for accuracy and completeness. 2 Points Total Points 8 Points **Standard and Target for Success:** Students scoring 70% or higher would have met a successful completion standard for CSCI 2. **Related Documents:** Fall 2014 SLO Report for CSCI 2.docx	**Semester and Year Assessment Conducted:** 2014-15 (Fall 2014) **Standard Met? :** Standard Met Results Number of students assessed: 41 Table below gives score distributions. Score percentage or range Number of students in that range Percentage of Students in each range (%) 100% 9 22 90 % to <100% 13 32 80% to <90% 9 22 70% to <80% 5 12 60% to <70% 1 2 50% to < 60% 1 2 <50% 3 8 Total 41 100 Interpretation of results 88% of the students successfully completed the assessment. For the students who met the target, I think they communicated well with the instructor, understood class lectures, studied the supporting materials and learned overall art of developing programs including use of data structures using C++. Instructors experience in teaching CSCI 2 may also have added to student engagement in the class, thus students being productive learners. Twelve percent of students however did not meet the course completion standards. That could have been due to combination of factors. Typical factors we have seen hindering student success in community colleges are:	**Action:** To support and improve student success, the study materials such as multimedia tools, Powerpoints, and PDF documents on hard to understand topics will be created and will be provided to all professors teaching CSCI 2. The multimedia materials such as videos are hosted on Satish Singhal youtube channel whose link will be provided to all students. (09/01/2015) **Action Category:** Teaching Strategies **Follow-Up:** At the beginning of fall 2015, all CS2 instructors were given following teaching and multimedia materials to integrate in their classes as needed. 1. Link to all singhal videos on C++ hosted on youtube. 2. All singhal E-book chapters in PDF and microsoft word were provided at a central El Camino server from where students can access them as needed. (11/19/2015)

Course SLOs	Assessment Method Description	Results	Actions
		1. Lack of engagement. 2. Demanding work and college schedule. 3. Borderline success in pre-requisite class or having done such class so long ago that due to lack of use the pre-requisite material has been forgotten. 4. Sudden change in student's life condition that required attention and time resources to be redirected from studies towards resolution of such condition.	
	Project - Here is a C++ class definition for an abstract data type LinkedList of strings. Implement each member function in the class below. Some of the functions we may have already done in lecture, that's fine, try to do those first without looking at your notes. You may add whatever private data members or private member functions you want to this class. #include <iostream> #include <string> using namespace std; using ItemType = string;	(02/12/2015) **Faculty Assessment Leader:** Satish Singhal **Faculty Contributing to Assessment:** Satish Singhal and Joe Hyman **Semester and Year Assessment Conducted:** 2018-19 (Fall 2018) **Standard Met? :** Standard Met Number of students assessed: 39 Table below gives score distributions. Score percentage or range Number of students in that range Percentage of Students in each range (%) 70% to 100% 31 79.49 50% to <70% 8 20.51 <50% 0 0 The table below shows the statistics of above data: Statistical Property Value of Statistical Property Highest = 100%	**Action:** Give these students some extra credit to research a topic on their own. (03/02/2020) **Action Category:** Teaching Strategies **Action:** Mention more advanced Data Structures (Tries, AVL trees, Red Black trees) (03/02/2020) **Action Category:** Teaching Strategies **Action:** Invite Guest Speakers to our classrooms. (03/02/2020) **Action Category:** Teaching Strategies

Course SLOs	Assessment Method Description	Results	Actions

	struct Node {
	ItemType value;
	Node *next;
	};
	
	class LinkedList {
	
	private:
	Node *head;
	
	public:
	
	// default constructor
	LinkedList() : head(nullptr) { }
	
	// copy constructor
	LinkedList(const LinkedList& rhs);
	
	// Destroys all the dynamically allocated memory
	// in the list.
	~LinkedList();
	
	// assignment operator
	const LinkedList& operator=(const LinkedList& rhs);
	
	// Inserts val at the front of the list
	void insertToFront(const ItemType &val);
	
	// Prints the LinkedList
	void printList() const;
	
	// Sets item to the value at position i in this
	// LinkedList and return true,
		Average = 78.8%
		Median = 80.0%
		Standard Deviation = 15.7%
		Lowest = 50%
		
		The analysis of data shows that SLO criteria of 70% student passing with 70% or higher score was met. This is because almost 79.49% student scored in the range 70 % or more. It would appear that students met SLO #1 rather robustly.
		
		In the light of such successful results for SLO #1 we propose that perhaps in next SLO #1 assessment, students may be asked to perform more difficult operations on data structures like linked-lists. There might also be the need to assess students on more advanced data structures like doubly-linked lists, and self balancing binary trees. We do realize that about 20 % of the students did not meet the SLO criterion. The reason for that may be:
		1. Lack of engagement.
		2. Demanding work and college schedule.
		3. Borderline success in pre-requisite class or having done such class so long ago that due to lack of use the pre-requisite material has been forgotten.
		4. Sudden change in student's life condition that required attention and time resources to be redirected from studies towards resolution of such condition. (03/01/2019)
		% of Success for this SLO: 79.49
		Faculty Assessment Leader: Solomon L Russell
		Faculty Contributing to Assessment: Solomon Russell, Edwin Ambrosio, Hwang Yih-Yu	

Course SLOs	Assessment Method Description	Results	Actions
	returns false if // there is no **Standard and Target for Success:** It is expected that 70% of students will score 70% or higher on this SLO for CSCI 2.		

ECC: CSCI 3:Computer Programming in Java

Course SLOs	*Assessment Method Description*	*Results*	*Actions*
SLO #1 Designing, Coding, Compiling and Testing - Students, when given a specification for a program or program segment, will be able to design, code, compile, test and document a solution. **Course SLO Status:** Active **Course SLO Assessment Cycle:** 2014-15 (Fall 2014), 2018-19 (Fall 2018) **Input Date:** 11/19/2013	**Project -** The students will be assigned a project to develop a full program from specification to final demonstration. **Standard and Target for Success:** It is expected that 85% of students will score 75% or above on this SLO. **Related Documents:** CS 3 SLO #1_Assessment Question for Fall 2014.docx	**Semester and Year Assessment Conducted:** 2014-15 (Fall 2014) **Standard Met? :** Standard Met Of 61 students assessed, 55 scored 75% or above on the project. That is a 90% success rate. This project was the 2nd of 7 projects in the course and was due during the fourth week of the 16 week term. All 6 of the unsuccessful students ended up withdrawing from the course. (02/02/2015) **Faculty Assessment Leader:** Gregory L Scott **Faculty Contributing to Assessment:** Esmaail Nikjeh	**Action:** Will review the effectiveness of this assessment tool/method and the relevance of the SLO with faculty. (01/22/2018) **Action Category:** SLO/PLO Assessment Process **Follow-Up:** This assessment is still effective and relevant. (03/14/2018)
	Project - The students will be assigned a project to develop a full program from specification to final demonstration. **Standard and Target for Success:** It is expected that 85% of the students will score 75% or above on this SLO.	**Semester and Year Assessment Conducted:** 2018-19 (Fall 2018) **Standard Met? :** Standard Met Of 52 students assessed, 48 scored 75% or above on the project. That is a 92% success rate. This project was the 2nd of 7 projects in the course and was due during the fourth week of the 16 week term. All 4 of the unsuccessful students ended up withdrawing from the course. (02/28/2019) **% of Success for this SLO:** 92 **Faculty Assessment Leader:** Edwin Ambrosio **Faculty Contributing to Assessment:** Gregory Scott, Esmaail Nikjeh **Related Documents:** CS3__0156_SLO_SCORES_E_NIKJEH_F18.xlsx CS3-0158and01590-FallSLOResults.xlsx	**Action:** Will review the effectiveness of this assessment tool/method and the relevance of the SLO with faculty. (02/28/2020) **Action Category:** SLO/PLO Assessment Process