

## Syllabus

EECE 135: Algorithms and Programs for Engineers

CSCI 111: Programming and Algorithms I

Instructor: Shaun McCluskey

email: shaunmcc@gmail.com

### Registration/Schedule Information:

Class Number	Section	ACT	Days	Time	Room	Instructor
	EECE-135-01	DIS	MWF	9:00-9:50	OCNL 254	
3801	EECE-135-02	ACT	M	10:00-10:50	OCNL 244	Shaun McCluskey
3843	EECE-135-03	ACT	W	10:00-10:50	OCNL 244	
3844	EECE-135-04	ACT	F	10:00-10:50	OCNL 244	

### Prerequisites:

At least one year of high school algebra is recommended.

### Required Text:

*C++ How to Program 6/e*  
H.M. Deitel & P.J. Deitel, 2008



### Additional Requirements:

We will be using *Clickers* (Student Response System)

See Dr. Juliano's Clicker Page:

<http://www.ecst.csuchico.edu/~juliano/csci111/>

All students must set up a College of ECST Unix account. See Elbert Chen in OCNL 249 Mon-Fri 3:30-5:00 to set one up. If that isn't possible contact him at [echan@csuchico.edu](mailto:echan@csuchico.edu).

WebCT Vista will be used, so make sure you have access.

### Course Objectives:

1. Help students learn and adopt a disciplined approach to algorithm design and implementation, programming, and widely accepted software engineering methods.

2. Facilitate students' understanding of basic principles and techniques in the fast-maturing field of computer science.
3. Teach students the proper use of programming tools to develop correct, efficient, and well-structured programs and to build a foundation for learning more advanced programming concepts and paradigms.

**Course Outcomes:**

1. Use a disciplined approach to algorithm design and implementation, programming, and widely accepted software engineering methods.
2. Demonstrate proficiency in the basic principles and techniques in the fast-maturing field of computer science.
3. Demonstrate proper use of programming tools to develop correct, efficient, and well structured programs illustrative of a foundation for learning more advanced programming concepts and paradigms.

**Grading Standards:**

<b>Class Participation/Attendance(10%)</b>	
100%	Clicker Participation
<b>Theoretical Component(45%)</b>	
20%	Exam 1
20%	Exam 2
20%	Exam 3
40%	Final
<b>Practical Component(45%)</b>	
100%	Programming Assignments

Students are required to earn a C- or better in **all three** grading components, otherwise I reserve the right to use the *minimum* of the three components to calculate the student's final grade. For example, if a student averages 90 percent on the practical component, and a 100 percent on class participation/attendance component, but averages only 65 percent on the theoretical component, the student can receive a final grade of 65.

**Final Grades:**

Raw Score	Letter Grade	University definition
[100-93]	A	Superior Work
[90-93)	A-	
[87-90)	B+	
[83-87)	B	Very Good Work
[80-83)	B-	
[77-80)	C+	
[73-77)	C	Adequate Work
[70-73)	C-	
[65-70)	D+	Minimally Acceptable Work
[60-65)	D	
[0-60)	F	Unacceptable Work

**Programming Assignments:**

There is a no collaboration policy for this class. Students are expected to submit their own solutions to all of their programming assignments.

**WARNING:** Don't even THINK about cheating in this class! Each semester, several students in this class either earn a final grade of F or are recommended for expulsion from Chico State for violating the course's *no collaboration* policy. This is not because CSCI 111 / EECE 135 students are any less ethical or honest than students in other classes; rather, it is because I will be using using MOSS (Measure Of Software Similarity, see <http://www.cs.berkeley.edu/~aiken/moss.html>) to automatically check for similar programming assignment solutions. MOSS is an Internet service that allows instructors to submit a set of programs for comparison. It is used by more than 1,200 users worldwide -- representing both individual instructors and entire departments. Each semester, several students think the use of MOSS is a bluff or that they can outsmart MOSS --- they are always proven wrong. (See <http://www.ecst.csuchico.edu/~juliano/csci111/Lab/MOSS/> for sample output provided by MOSS.) *Consider yourself warned.*

**General Policies:**

Late programming assignments will not be accepted. Each assignment will have its due date posted well enough ahead of time for students to plan their schedules. I have learned through experience that allowing late assignments puts an undue burden on me and the

students who turn in assignments on time. The three midterms and the final will be given during class periods. No late exams will be given. If a student has a compelling reason to miss an exam, I need to be notified prior to the exam date. Granted there may be circumstances beyond a student's control, but these need to be grievous issues and will be dealt with on a case by case basis.

### **Tentative Schedule:**

<b>Week</b>	<b>Chapter</b>	<b>Coverage/Comments</b>
1	1	Introduction to Computers, the Internet, and World Wide Web
2	2	Introduction to C++ Programming
	3	Introduction to Classes and Objects
3	3	Introduction to Classes and Objects, <i>continued</i>
	4	Control Statements Part 1
4	4	<b>Exam 1</b> , Control Statements Part 1, <i>continued</i>
5	5	Control Statements Part 2
6	6	Functions and an Introduction to Recursion
7	6	Functions and an Introduction to Recursion, <i>continued</i>
	7	Arrays and Vectors plus: MSWin Project Files and *nix Makefiles
8	7	<b>Exam 2</b> , Arrays and Vectors, <i>continued</i>
9	8	Pointers and Pointer Based Strings, Command Line Arguments
10	9	Classes, a Deeper Look, Part 1
11	10	Classes, a Deeper Look, Part 2
	11	Operator Overloading; String and Array Objects
12	11	Operator Overloading; String and Array Objects, <i>continued</i> <b>Exam 3</b>
13	12	Object Oriented Programming: Inheritance
14	13	Object Oriented Programming: Polymorphism
15	19	Searching and Sorting
16		<b>Final Exam</b>