

## CS102 / ECE206 An Introduction to Computer Science

**Credits and Contact Hours:** 4 credits, 2.5 lecture hrs. / week, 3 lab hours / week

**Instructor's or Course Coordinator's Name:** Bruce MacLennan

### Textbook and Other Supplemental Material

- a. *Learning Computing with Robots in C++*, ed. by Deepak Kumar, 2010.
- b. *How to Think Like a Computer Scientist: Learning with C++*, by Allen B. Downey, 2010.
- c. online notes and example programs.

### Specific Course Information

- a. **Catalog Description:** Problem solving and algorithm development. Organization and characteristics of modern digital computers with emphasis on software engineering, building abstractions with procedures and data, and programming in a modern computer language. Includes Level 1 design projects, which require laboratory work.
- b. **Prerequisites and/or Co-requisites:** none
- c. **Required**

### Specific Goals for the Course

- a. specific outcomes of instruction:
  - (1) Students will gain experience in algorithmic problem solving.
  - (2) Students will be able to understand and apply basic programming techniques.
  - (3) Students will understand and apply basic approaches to program development.
  - (4) Students will be able to read and write basic C++ programs.
- b. Student Outcomes listed in Criterion 3

Course Outcomes	Engineering Accreditation Commission Statement of Student Outcomes												
	a	b	c	d	e	f	g	h	i	j	k	l#1	l#2
alg. problem solving			X		X								
basic prog. techniques			X								X		
program development			X								X		
C++ prog.											X		

or:

Computing Accreditation Commission Statement of Student Outcomes											
Course Outcomes	a	b	c	d	e	f	g	h	i	j	k
alg. problem solving		X	X							X	
basic prog. techniques			X						X		X
program development			X						X		X
C++ prog.									X		

### Topics Covered

**basic programming concepts** (arithmetic, boolean logic, assignment, conditional execution, iteration, input-output, function definition, recursion, pseudo-random numbers, pointers, dynamic memory management, linked lists), **C++ syntax and semantics** (numbers, characters, strings, expressions, function definition, if-else, switch, for, while, <<, >>, file I/O, formatted I/O, strings, vectors, structs, enumerations, typedef, pass by value, reference, and constant reference; STL: vectors, lists), **object-oriented programming** (classes, member functions, constructors, subclasses and inheritance, public, private, protected, virtual definition, overloading, templates), **program development** (top-down, bottom-up, incremental, modular design, unit testing, input validity checking, preconditions, invariants and assertions), **sorting and searching** (bubble, selection, mergesort, sequential search, bisection search), **robotics and AI** (Braitenberg vehicles, reactive control, behavior-based control, game playing, look ahead).