Required

CSE 142 Computer Programming

Credits
4.0, (3 hrs lecture, 1 hr section)

Lead Instructor
Stuart Reges

Textbook
- *Building Java Programs: A Back to Basics Approach*, Reges & Stepp

Course Description
Basic programming-in-the-small abilities and concepts including procedural programming (methods, parameters, return values), basic control structures (sequence, if/else, for loop, while loop), file processing, arrays and an introduction to defining objects.

Prerequisites
None

CE Major Status
Required

Course Objectives
Students will master basic procedural programming constructs. They will learn to write and debug small programs (50-100 lines) using a full range of procedural techniques using a variety of input sources (console, file) and a variety of output destinations (console, file, graphical). Students will learn the design principles that are relevant to this style of "programming in the small" including decomposition, information hiding, elimination of redundancy, detailed documentation, and use of parameters and return values to create flexible components. Students will be exposed to the major design issues of object oriented programming including inheritance.

ABET Outcomes
(1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
(2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, , and economic factors (H)
(7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies (H)

Course Topics

- Basic concepts of computer structure and program execution
- Variables, types, expressions, and assignment
- Input/output: console, file, graphical
- Conditional execution (if/else)
- Iteration (for, while)
- Defining methods: parameters, return values
- Arrays (one-dimensional)
- User-defined classes
- Use of standard library objects: strings, graphics
- Procedural decomposition of problems
- Programming style: eliminating redundancy, localizing variables, class constants, commenting, use
- of parameters and return values to increase flexibility, appropriate choice of control structure
- (e.g., sequential if versus if/else)