EECE 230: Introduction to Programming

Catalog description:
A course on the basic principles of programming and their application to the solution of engineering problems using a high level programming language. This course introduces structured and object-oriented programming, and covers the basic data types, control structures, functions, arrays, pointers, and classes. Weekly laboratory assignments are an integral part of this course.

Credit hours: 3 credits

Required or elective:
Required for CCE / ECE students

Prerequisites:
By course: No Prerequisite
By topic: Elementary calculus and arithmetic’s

Textbook(s) and/or required materials
Textbook: C++ Programming: Program Design Including Data Structures, by D.S.Malik, THOMSON Course Technology. Additional material will be placed on reserve in the library and posted on Moodle.

References:
None

Course Objectives
1. Introducing students to Computer Programming.
2. Basic principles of structured and object-oriented programming using a high level language.
3. Basic techniques of formulating problems for computer programming implementation and solution.
4. Practical computer programming skills through the solution of engineering problems using the C++ high-level

Course Topics
1. Introduction to programming
2. Data Types and Programming Basics
3. Control structures
4. Functions
5. Arrays and Strings
6. Applications to Searching and Sorting
7. Classes and Data Abstraction
8. Pointers

Course Learning Outcomes
1. Can edit, compile, and run a program using the C++ integrated development environment.
2. Know the various built-in data types in the C++ programming language.
3. Know how to input/output data (cin, cout, files…)
4. Know how to use the C++ control structures (for, while, if/else…).
5. Know how to modularize a program through the use of functions.
6. Know how to write functions with call by value and call by reference.
7. Know how to write recursive functions.
8. Can use arrays in programs and functions.
9. Have learned and used the string manipulation functions.
10. Know how to implement data abstraction through the use of Classes: public and private members, member functions, constructors, destructors.
11. Know how to use pointers and dynamic memory allocation in programs.
12. Have developed basic knowledge of array-based searching and sorting algorithms.
13. Have gained hands-on experience in C++ programming through weekly lab exercises and assignments.
14. Know how to apply the principles of computer programming to the solution of engineering problems.
15. Value issues in Computer Ethics

Class/laboratory schedule
a- Three 50-minute lectures per week
b- One 3-hours lab session per week
c- Use of computer lab is needed for working on the assignments.

Resources of the course
Textbook, C++ compiler, and Moodle.

Computer usage
Programming in C++ language.

Evaluation methods
- Programming Assignments - to be mostly done in the lab - 10%
- Quiz-I – programming 25%
- Quiz-II – programming 30%
- Final Programming 35%
- In case of major discrepancy between lab and exam performance, the lab grade will be discarded
- If the student exhibits progressive improvement in the course, the above percentages might be changed
Professional component
Engineering topics: 75%
General education: 0%
Mathematics and basic sciences: 25%

Person(s) who prepared this description and date of preparation
Ali El Hajj, February 2007

Date of last revision
February 2013