

ECE 131 – Programming Fundamentals

Module 1, Lecture 1: Course Administration/Overview

Dr. Daryl Lee

University of New Mexico



Module Overview

- Course Administration
- Review of the syllabus
- Goals of the course
- Course overview
- Background

Course Administration

Professor: Daryl Lee
daryllee@ece.unm.edu

Course: The class consists of this section only.

Courseware: Course materials are available through WebCT at:
vista.unm.edu.

Syllabus Review

Please read the course syllabus, and make sure you understand:

- dates and format of exams,
- homework and programming assignments,
- course policies,
- teaching assistants,
- required materials,
- etc.

DO YOU HAVE ANY QUESTIONS ABOUT HOW THE COURSE
WILL BE ADMINISTERED?

Course Goals

- To provide a basic introduction to programming that gives the student an appreciation for the scope of modern programming activities, and provides a solid foundation for further study in this area:
 - A brief history of computing, basic programming models and programming paradigms.
 - An understanding of how programs are made (the edit-compile-link-execute cycle), prototyping and project organization.
 - Basic programming elements – numbers and their representation, program flow and control structures.
 - Pointers and arrays – indirect addressing, pointer arithmetic and array indexing, memory management.
 - Functions and subroutines – separate compilation, formal vs. actual parameters, call-by-reference and call-by-value.
 - How computers interact with the external world – File I/O, bit-level operations and computer interfacing, embedded computing.

Background

- Computing technology has now eclipsed manufacturing as the dominant driver of the economy worldwide.
- Businesses are now utterly dependent on software products for every aspect of their operation, including hiring, management, planning, marketing, sales and investment.
- Software is now a component in nearly every manufactured product that has any level of complexity. This software often runs on a special-purpose computer *embedded* in the product:
 - Consumer Electronics – cell phones, watches, mp3 players and other consumer devices have been controlled by embedded computers for quite some time.
 - Automobiles – embedded computers now commonly used to control the transmission, safety equipment, and traction.
 - Appliances – the operation of everything from ovens to vending machines is now commonly controlled by embedded computers.
 - Industrial Machinery – robots, cooling equipment, assembly lines, etc. are all under computer control in a modern plant.

Background

- Every type of engineer should be prepared to deal with the creation of software products:
 - A Matlab simulation of a physical process.
 - A spreadsheet macro for evaluating an equation.
 - A database script for storing and tracking test results.
 - User-interface code for a product.
 - Software for controlling the operation of the same product.
- For many years, companies have struggled to develop software that works properly — and there have been many failures, including some that involve loss of life.
- The ability to use software to solve problems and create opportunities has improved dramatically over the past few years, as software tools have become much better, and frameworks (e.g., the Internet) have emerged.
- Nevertheless the “software age” is in its infancy, and you can expect many changes in how software is created and used in the coming years.
- Thus, it is critical to build a solid foundation for understanding software that will maintain its relevancy in the future — and that’s what we intend to do in this course.
- So let's get started!

Background

- Your first homework assignment:
 - Choose a work environment (OS X, Linux, Windows).
 - Create a text file named `yourUserName.txt`.
 - Include the first line of output from “`gcc -version`”.
 - Add a line saying what your work environment is.
 - Email it to me as an attachment.
 - If you are using Windows and want a local environment:
 - <http://sourceforge.net/projects/mingw/files/Automated%20MinGW%20Installer/mingw-get-inst/>
- Due Date: One week from today.