UNIVERSITY OF CALIFORNIA, SANTA CRUZ
BOARD OF STUDIES IN COMPUTER ENGINEERING

CMPE-13/L: COMPUTER SYSTEMS AND C PROGRAMMING

SPRING 2011
M-W 5:00 – 6:45 PM
CLASS: Stevensen Academic Room #175
LAB: Ming Ong Computer Library, 108

INTRODUCTION

Computer Systems and C Programming is a class intended to bring you up to speed on programming small and large programs in C. Originally written in 1978, C remains the most popular programming language, and the most used one in terms of numbers of computer programs written in it. There are no prerequisites, but you are expected to be at least somewhat familiar with programming and computers. In this class, we are going to approach C from an embedded paradigm, and all of your programming assignments are going to be on a 16-bit embedded micro, the Microchip PIC24 (or dsPIC33). You will learn how to program in C, how to write modular code, and some of the tips and tricks when dealing with an embedded micro. This is a programming class and you will be writing lots of code. Expect to spend at least 15-20 hours outside of class playing with the code to get things to work.

INSTRUCTOR:

GABRIEL HUGH ELKAIM
E-mail: elkaim@soe.ucsc.edu
Office: Engineering 2 (E2), 337B
Hours: M-W 2:00-4:00 PM
Phone: (831) 459-3054 (Office)
        (831) 425-3678 (Home: 10AM to 9PM please)

TAS AND HELPERS::

Bryant Mairs: bmairs@ucsc.edu
Nathan King: roguelikedev@gmail.com
Shelby Thomas: sthomas1@ucsc.edu

E2-316 (Autonomous Systems Lab), 9-2140

TEXTBOOKS (ALSO IN THE LIBRARY ON RESERVE):

THE C PROGRAMMING LANGUAGE

[Notes]: “Notes to accompany K&R,” by Steve Summit available on the class website and at: http://www.eskimo.com/~scs/cclass/krnotes/top.html


Optional Textbooks (good references):


READINGS

There is quite a bit of material to cover in this class, and you are expected to have read the assigned reading before coming to class. You will get out of this class what you put into it. Simply put, if you do not do the reading, you will not effectively learn the material. We have gone to quite a bit of trouble in order to find appropriate reading for you, so take the time to read them. The main textbook, The C Programming Language, (K&R) is one of the best books you will ever read on a technical subject. Like the language C itself, it is precise and to the point, and assumes you know what you are doing. There is quite a bit of subtle hints on how to code well in the book, and it should be read carefully. The Notes to Accompany K&R point out some of these subtleties and expand on some of the examples.

DEVELOPMENT ENVIRONMENT

We are going to be using a Microchip 16-Bit Processor, the PIC24, for all of the programming assignments in this class. All of our development will be done within the MPLAB integrated development environment.
development environment (IDE), which can be downloaded from the class website (see the Installing Software handout). We are using the Microchip C-30 compiler for PIC24 and dsPIC. Lastly, we will be using both a small development kit, the MicroStick, and a full blown simulator called the Proteus VSM, which simulates a more expensive development kit (the Explorer16) and a host of lab equipment.

Except for the MicroStick, all of these are available for free, and can be downloaded from the links on the class website. The MicroStick is available for $25, and Microchip offers a 25% discount for students. In order to communicate with the MicroStick, you will also need a USB to serial converter (Sparkfun makes one that we recommend for $15). Details will be on the website.

**Grading**

This course is based on a combination of the lab and class. They go together and are indivisible (they are divided only because that is how things are done here, ideally it would be just one class). If you cannot complete the lab, you cannot complete the course. You will receive the same grade on the course and the lab, and cannot pass one without passing the other. This is programming class, and you will be graded largely on the programs you write.

<table>
<thead>
<tr>
<th>COURSE/LAB:</th>
<th>60%</th>
<th>Programming assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40%</td>
<td>In-class quizzes (every week on reading)</td>
</tr>
</tbody>
</table>

Note: there is a sure fire quick and easy way to fail this class—that is to wait until the day before the programming assignment is due to start it.

**Note on Attendance: CMPE-13/L**

There will be a quiz at the beginning of class covering the required reading material *at least once a week*. No make-up quizzes will be given, nor will there be an opportunity to take the quiz later in that class. If you must miss a quiz, contact the instructor or TAs *before* the class to make arrangements. The only accepted excuse after a quiz is from the ER.

**WWW Site, Videos, and WebForum**

**Website:** www.soe.ucsc.edu/classes/cmpe013/Spring11

Check this site often as this is where the lecture notes, labs, quizzes and quiz solutions, and lecture videos are posted. You are expected to read the material on the website.

**Videos:** As an experiment in teaching technology, the instructor will be capturing both the audio and screen from the course in real-time. This will later be posted to the website. While every attempt will be made to capture the classes, as this is a new technology, there are no guarantees. Also, while watching the video should be a great way to review course material, if you are not in class you cannot ask questions and/or clarifications. Further, if too few students show up for lectures, the instructor may restrict access to class videos.

Use the webforum to post questions to the tutors and the TAs about lab and class work. Use it to ask questions of other students. Do not expect quick replies from the instructor, use e-mail for that. Do NOT post code onto the webforum.

COURSE WORK: CMPE 13

Attendance is highly recommended for the lectures as the material builds up quickly. Lecture material will be made available on the website, usually before covered in class. Annotated lecture notes and videos of the lecture will be posted after class in a timely fashion (see note on video above).

There will be weekly quizzes at the beginning of class that are both required and graded. They are worth 40% of your overall grade, and they are essential to mastering the material. In my experience, understanding the material requires at least three passes—(1) reading the material before lecture, (2) attending lecture and paying attention, and (3) synthesizing the material into your own work during the programming assignments. Skipping any of the steps requires a whole lot more work to make it up on your own.

LAB WORK: CMPE 13 L

This class should be a lot of fun. I would add also a lot of work (think: drinking from the firehose). There is a lot to cover, and only 10 weeks to get you familiar and confident in programming in C. Prepare to spend over 20 hours a week on this class, less at first, much more towards the end. We will make every attempt to help you, and to ensure that you succeed, but you have to put in the work yourselves.

You must be enrolled in CMPE 13L to remain in this class. You must pass CMPE 13 to pass CMPE 13L. You will receive the same grade for both the lab and the class, and thus cannot pass one without passing the other. In addition to the class lectures, you are expected to attend lab section twice weekly. In addition to going to the lab sections, you are free to go to additional lab sections as space allows.

We will be working with a 16 bit microcontroller, the MicroChip PIC24, using an actual microcontroller as well as a full simulator. See the section on Development Environment above for details about the tool chain. It is strongly recommended that you install the full development environment on your own computer. You will be working on it quite a bit, and it is nice to have your own place to do it in.

As a programming class, you are going to be doing a lot of programming. The bulk of your grade is based on the programming labs. As such, you will be helping each other out in the labs to understand the material. However, this is NOT license to copy others’ work. Credit for collaboration should be explicitly noted; failure to give credit on collaboration is considered a form of cheating and will be dealt with accordingly.
ACADEMIC HONESTY

Academic honesty is a requirement for the course. All assignments must be your own independent work; this includes quizzes, programming assignments, and labs.

What is cheating? It is presenting work that is not yours as your own. You can, and are encouraged to, discuss and strategize with your colleagues on the material and labs, but your work should be your own. Copying is NEVER acceptable.

On the labs, cheating is sharing code when not explicitly told that it is permitted. If a student is caught cheating in either the class or the lab this will result in an immediate failure in the class and the lab. It will be reported to your college and your department. DO NOT CHEAT; it is not worth it.

ACKNOWLEDGEMENTS

I would like to acknowledge Cyrus Bazeghi, for all his help with the course material, organization, and lecture material. Steve Summit, who taught an Introductory and Intermediate Programming class at the Experimental College at the University of Washington in Seattle, WA, has generously allowed us to use his notes on K&R and other supplementary materials. Microchip Corp. has generously provided slides and software through their academic partner program, and many of the slides come from Henry Cheng at UC Davis.