# CMPE-013/L Syllabus

## CMPE-13/L: Computer Systems and C Programming
### Spring 2014 Lecture/Lab Calendar


*Notes to accompany CMPE-13/L* by Steve Seuman available on the class website and also available at: [http://www.cs.kim.edu/~cs13/lnotes/psp.html](http://www.cs.kim.edu/~cs13/lnotes/psp.html) [Notes]

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Reading (to be completed before lecture)</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>K&amp;R: Chapter 1.1-1.2, 2.1, 4.3, 4.11.1 [Notes]: Same</td>
<td>Lab 1: Compiling, Running, and Debugging Due: 9-Apr-14 @ 11:59 PM Quiz 0 (K&amp;R:1.1-1.2)</td>
</tr>
<tr>
<td>2</td>
<td>K&amp;R: Chapter 2.10-2.11, 5.1, 5.7-5.8, 2.2, 2.7 [Notes]: Same</td>
<td>Lab 2: Simple Calculator Due: 10-Apr-14 @ 11:59 PM</td>
</tr>
<tr>
<td>3</td>
<td>K&amp;R: Chapter 3.5, 3.9, 4.7, 7.2, 7.4, 2.5-2.6 [Notes]: Same</td>
<td>Lab 3: Metric Math Library Due: 23-Apr-14 @ 11:59 PM</td>
</tr>
<tr>
<td>4</td>
<td>K&amp;R: 1.4-1.7, 1.5-1.7 [Notes]: Same</td>
<td>Lab 4: Randomness Due: 25-Apr-14 @ 11:59 PM</td>
</tr>
<tr>
<td>5</td>
<td>K&amp;R: Chapter 4.1, 4.2, 4.5, 4.8 [Notes]: Same</td>
<td>Lab 5: Linked List Due: 7-May-14 @ 11:59 PM</td>
</tr>
<tr>
<td>6</td>
<td>K&amp;R: Chapter 5.1-5.4, 5.6, 8.5 [Notes]: Same</td>
<td>Lab 6: Bounded Buffers Due: 14-May-14 @ 11:59 PM</td>
</tr>
<tr>
<td>7</td>
<td>K&amp;R: Chapter 6.1-6.2, 6.4, 6.5, 2.9, 2.12 [Notes]: Same</td>
<td>Lab 7: Toaster Due: 21-May-14 @ 11:59 PM</td>
</tr>
<tr>
<td>8</td>
<td>K&amp;R: Chapter 7.2, 7.7 [Notes]: Same</td>
<td>Lab 8: Binary Search Due: 21-May-14 @ 11:59 PM</td>
</tr>
<tr>
<td>9</td>
<td>K&amp;R: 5.6-5.9, 7.8, 8.5 [Notes]: Same</td>
<td>Lab 9: Interrupts Due: 21-May-14 @ 11:59 PM</td>
</tr>
<tr>
<td>10</td>
<td>Chapter 3.3-3.7 [Notes]: Same</td>
<td>Lab 10: Event-Driven Programming Due: 21-May-14 @ 11:59 PM</td>
</tr>
<tr>
<td>11</td>
<td>K&amp;R: Chapter 4.2, 4.3, 4.5, 4.11, [Notes]: Same</td>
<td>Lab 11: State Machines Due: 21-May-14 @ 11:59 PM</td>
</tr>
<tr>
<td>12</td>
<td>K&amp;R: Chapter 5.6-5.10 [Notes]: Same</td>
<td>Lab 12: State Machines Due: 21-May-14 @ 11:59 PM</td>
</tr>
</tbody>
</table>
CMPE-013/L Piazza

University of California, Santa Cruz - Spring 2013
CMPE 13/L: Computer Systems and C Programming

**Description**

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 (CMPE12 is recommended).

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.

**General Information**

Lecture Information:
TTh 2:00 – 3:45 PM
CLASS: Physical Sciences 114
LAB: Jack Baskin Engineering, 109
Mem Org Lab, 108

Textbooks:

*Notes*: ‘Notes to accompany K&R’ by Steve Summit available on the class website and at: http://www.cs.ucsc.edu/~sknotes/top.html

Optional Textbooks (good references):


CMPE-013/L Books


- **[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

CMPE-013/L Optional Books


CMPE-013/L Optional Books


CMPE-013/L Attendance

• Highly Recommended
  – Material builds up quickly
  – Videos available, but not the same as being there

• In class quizzes, beginning of class on Thursdays
  – No excuses accepted post-quiz.

• Be able to attend least one lab section per week (BE109)
  – Easier to get through the labs with help
  – TA/Tutors will be available then
  – Space reserved for your “signed up” section
Academic Honesty

• Cheating is presenting someone else’s work as your own
• All code turned in will be run against a code-checker
• Anyone caught cheating will immediately fail the class and the lab, and be reported to their college
• Copying each other’s code is never acceptable.
• Don’t do it—not worth it.
CMPE-013/L Grading

• Lecture and Lab are one and the same: CMPE-013 and CMPE-013L will get the same grade, same evaluation.

• In class quizzes (once per week): 30%
• Programming assignments (one per week): 70%

• No midterm, no final
CMPE-013/L Lab Work

• All programs we are using can be loaded onto your own laptop for use at any time—they are all free.
• We’re using microchip’s MPLABX IDE and XC32 compiler
• Running on a microcontroller development board by Digilent, the Uno32
• You can buy the this hardware directly from Microchip if you want to after the class is over
Lab Kit

Uno32

Basic I/O Shield

PICkit3
Reading Assignment for Today

• [Gonick]: parts 1-3, review your notes from CE-12/L
  – There will be a quiz on this on Thursday
• Check out the Webpage and Piazza Forum
Reading Assignment for Thursday

- **[Gonick]:** parts 1-3, review your notes from CE-12/L
  - There will be a quiz on this on Thursday
- Lab 0 Documentation
- **[K & R]:** Chapter 1.1-1.2, 2.1, 4.5, 4.11.1
Lab 0: “Compiling, Running, and Debugging”

- Due on Wednesday, 9-Apr-2013 at 11:59PM
- Get started early
- Submit early!
- Make sure to read documentation
- Code submitted electronically using submit command through UNIX shell.
- Full lab on Website
What is a Computer?

• Computer
  – A computer is a machine that manipulates data based on a list of instructions called a program.
  – A computer consists of hardware and software.

• Computer Hardware
  – Computer hardware is the physical part of a computer.
  – A typical computer consists of central processing unit (CPU), main memory and external memory, and input and output devices.
  – A CPU consists of control unit (CU), arithmetic and logic unit (ALU), and registers.
The von Newmann Computer Architecture

Both programs and data are stored in the same memory
The Harvard Computer Architecture

Programs and data are stored in different memory
Programming Languages

Many differences including:

1. Abstraction type
2. Compiled vs. interpreted
3. Memory management
4. Type system
The C Programming Language

• Procedural
• Compiled
• Manual memory management
• Statically typed
• Small
• Low overhead
Comparison of C with Machine Code and Assembly Code

The memory addresses, machine code, and assembly code corresponding to a C statement \( a = a + 8 \) for the Rabbit 3000 8-bit microprocessor.

<table>
<thead>
<tr>
<th>Memory address</th>
<th>Machine code</th>
<th>Assembly code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0X1EA1</td>
<td>000100010000100000000000000000</td>
<td>ld de,0x0008</td>
</tr>
<tr>
<td>0X1EA4</td>
<td>110001000000000000000000000000</td>
<td>ld hl,(sp+0)</td>
</tr>
<tr>
<td>0X1EA6</td>
<td>00011001</td>
<td>add hl,de</td>
</tr>
<tr>
<td>0X1EA7</td>
<td>110101000000000000000000000000</td>
<td>ld (sp+0),hl</td>
</tr>
</tbody>
</table>
Reasons to Learn C


1. C is one of foundations for modern information technology and computer science.
2. C is the most commonly used programming language in industry.
3. C is a standardized programming language with international standards.
4. C is available for almost every computing platform that is currently in production.
5. C is the language of choice for programming embedded systems.
6. Once you have learned C, you can pick up other languages without much difficulty by yourself because all other modern languages borrowed heavily from C.
History of C

• **C**
  – Invented by Ritchie based on B, a simplified version of BCPL
  – Used to develop the Unix operating system and Unix commands
  – Most system software such as OS are written in C or C++
  – Replacement for assembly language for hardware interfacing
  – By 1978 C had evolved to “K & R C”

• **C Standards**
  – 1\textsuperscript{st} C standard created in 1989 by ANSI, ratified by ISO in 1990. It is called C89. Some call it C90.
  – 2\textsuperscript{nd} C standard was ratified in 1999, called C99.
  – 3\textsuperscript{rd} C standard ratified in 2011, called C11.
Just the Facts

- C is more "low level" than other high level languages (good for MCU programming)
- C is supported by compilers for almost all architectures
- C can do *almost* anything assembly language can do
- C is usually easier and faster for writing code than assembly language
What we will cover in 13/L

• “C” programming
  – Using C in an Embedded Environment
  – Comments
  – Variables, Identifiers, and Data Types
  – Literal Constants
  – Symbolic Constants
  – printf() / scanf()
  – Operators

  – Expressions and Statements
  – Conditionals
  – Loops
  – Functions
  – Multi-File Projects & Storage Class Specifiers
What we will cover in 13/L

• “C” programming (con’t) • Advanced Techniques
  – Arrays
  – Data Pointers
  – Function Pointers
  – Structures
  – Unions
  – Bit Fields
  – Enumerations
  – Macros with #define
  – State Machines
  – Recursion
  – Interrupts
  – Program decomposition
  – Abstraction
  – Scope
  – Static / Dynamic Memory allocation