CMSC 198K: The “Science” behind Computing
Fall 2009

Instructors: Samir Khuller and Bobby Bhattacharjee
E-mail: {samir,bobby}@cs.umd.edu
Phone: +1 301 40{5 6765, 5 1658}

Class meets: 9:30 a.m. – 10:45 a.m. Tuesdays and Thursdays, CSIC 1115
Office Hours:
Samir Khuller 11:00 a.m. – 12:00 p.m. Mondays and Thursdays, AVW 3369
Bobby Bhattacharjee 10:00 a.m. – 12:00 a.m Wednesday, AVW 4143
Kristin Stephens 2:00 p.m – 3:00 p.m. Tuesday and Thursday, AVW 1112

Class web site: http://www.cs.umd.edu/class/fall2009/cmsc198K

Course Overview
ENIAC (Electronic Numerical Integrator And Computer) is considered the first “general-purpose” electronic computer. It weighed 30 tons (literally), consumed as much power as a small hotel, and cost about $6M (in 2008 dollars). In 1946, when it was introduced, it was an amazing machine: it could do five thousand additions or subtractions every second, and was generally heralded as a “giant brain”.

Fast forward sixty years: today’s computers are thousands of times smaller, consume one thousandth the power, have reduced in cost by a factor of one thousand while (literally) being able to store billion times more information and processing it billion times faster.

No single technology in human history has undergone such a dramatic improvement (over any time scale) and yet, the perhaps the most surprising fact is not how much smaller, cheaper, or faster computer hardware has gotten, but the range of problems that computing technology efficiently addresses. Think about how much this technology is part of your daily life: Twitter and Facebook, IMs and e-mail, the Web and the Internet, credit cards and electronic banking, iPods and cell phones, XBoxes and Wiis, GPS and computer controlled automatic transmissions in your car, auto-pilots on airplane flights and computer-controlled electric power grids, . . . are all applications of this relatively young discipline. How are we able to use computers to put together millions of small strands of DNA read by machines into a large human genome? Modern society without computing technology is unimaginable to most of us.

How did this technology come about? Is there a common underpinning to all of these applications of computers? For that matter, how different are the computers that control an airplane vs. your iPod vs. your toothbrush? Do computers really “talk” over the Internet? What is the Internet? Who invented it? How does Google Maps know about the directions in your neighborhood? How do search engines find the one page you were looking for amongst the millions that matched? What is the “science” behind this incredible range of technologies?

Computer Science is the systematic study of computation and its applications. The computing technologies touched on above are based on a number of ideas in Computer Science. In this course, you will be introduced to some of these basic ideas in Computer Science, and learn how these ideas have been developed and are applied in everyday technology.

A fundamental concept in Computer Science is that of an algorithm. Algorithms describe how a computer should undertake a given task, e.g., Google uses sophisticated algorithms to select the best pages for a given search; your iPod uses compression algorithms to store your music and so on. Computer Scientists discover algorithms for solving problems and also analyze how well these algorithms work. Computer scientists have developed a framework for reasoning about the intrinsic complexity and speed of algorithms, which you will be introduced to in this course. Along the way, you will learn about many fundamental (and remarkably simple) algorithms that are used in programs and devices you use every day.

Programming is an expression of algorithms in a form that can be executed by a computer. Programming languages are much like human languages in that they are used to express algorithms. Like literature, good
programs are elegant expressions of beautiful ideas. In this course, you will learn the basic ideas behind computer programming and develop your own programs.

The Computer Systems component of this course will introduce you to how modern computer hardware and software is structured. You will learn about the common building blocks of modern computers, how system software is structured, and principles behind computer-to-computer communication. We will also cover the basics of computer architecture and rudimentary distributed algorithms that coordinate multiple computers.

In summary, the course will present an interleaved introduction to major ideas in algorithms, programming, and computer systems.

1 Reading

Currently, there is no single textbook that matches the material we plan to develop and cover. However, the following books do contain some of the material that we will cover.


- A book on Ruby.


The majority of the class will be taught using lecture notes that will be available on the course web page this Fall, and additional online resources.

2 Getting Help

There is a forum (forums.cs.umd.edu) for this course. Please use the forum to post questions and answers that may be useful to others. We will also update the class web site during the semester (add new papers etc.) You will be responsible for all announcements on the forum and the web site.

In general, post non-personal queries to the forum first. If the same question is posted to the newsgroup and sent via e-mail (possibly by different people), we will post answers to the forum.

Each instructor and TA has 2 hours of office hours every week. Our office hours are also posted on the class page. You are welcome to come by at other times after making an e-mail appointment. Please remember to put the string CMSC 198K: in the subject line of your e-mail. (Messages that have that string in their subject line are sure to be read).

3 Grading

The grading allocation is given below.

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homeworks</td>
<td>20%</td>
</tr>
<tr>
<td>Two exams</td>
<td>20% each</td>
</tr>
<tr>
<td>Project</td>
<td>10%</td>
</tr>
<tr>
<td>Final</td>
<td>30%</td>
</tr>
</tbody>
</table>

You will be responsible for all material covered and assigned in class.

It is your responsibility to make sure that you pick up all homeworks and handouts. All course information and homeworks will be available on the web page. Solutions to homeworks will be given out in class.
Syllabus

The topics and order listed below are tentative and subject to change.

1. Computing Technology is based on Computer Science
   Distinction between technology (timely) and science (timeless)

2. A Brief History of Computing [running theme]
   Major figures; big ideas

3. Digital Computers
   Roomba vs. Polaris
   Concept of stored program computing and the Church-Turing Hypothesis

4. What is an algorithm?
   Introduction by examples

5. What is a computer program?
   Programming by example

6. Applications of Algorithms
   Google News, Search, DNA Sequencing

7. Simple algorithms
   Stable marriages, GCD, primality, binary search

8. Programming fundamentals
   decisions, loops, program structure

9. Lifecycle of a program
   Code, compilers, and executables

10. What is inside your computer?
    High-level overview of computer hardware/components

11. Why is my computer slow?
    Architecture basics: Memory hierarchy, pipelining

12. Who is in charge?
    System software, scheduling policy, network stack

13. Sorting and Selection: Sorting techniques, Selection

14. Graphs and search techniques
    Depth first search, Breadth first search, Shortest Paths

15. Elementary Cryptography
    Classical ciphers, one time pads

16. Error Correction
    Parity
17. Networking and the Internet
   Basics of computer protocols

18. Analysis of Algorithms:
   Analysis of algorithms, worst case behavior, Order notation, Asymptotics.

19. Complexity of Problems
   Intrinsic complexity, idea of complexity classes

4 Policy and Academic Honesty

- Please turn assignments in on time. Unless previously negotiated, you will receive no credit for work that is not turned in on the day and time it is due. The only exception is for excused absences as defined by the university (Section V-1.00(G) of the Consolidated USMH & UMCP Policies and Procedures Manual).

- Do not miss exams. Unless previously negotiated, you will receive zero credit for missed exams. Once again, the only exception is for excused absences as defined by the university.

- Do not cheat. Plagiarism will not be tolerated. Please read and understand the UMCP code on academic integrity (Section III-1.00(A) of the Consolidated USMH & UMCP Policies and Procedures Manual http://www.inform.umd.edu/CampusInfo/Departments/PRES/policies/iii100a.html). Do not violate it. It is not worth your time (or mine) to be here if you do.