



Computer Science at Berkeley



Merry Choi
12/03/07



Lower Division CS Courses

- 4 lower division CS courses required for EECS majors:
 - 61A: Fundamentals of Computer Science
 - 61B: Data Structures
 - 61C: Machine Structures
 - 70: Discrete Mathematics and Probability Theory
- + various math and EE courses



CS61A: Fundamentals

- Recursion (you've got it!)
- Object-oriented programming
- Other areas of computer science:
 - Efficiency
 - Synchrony
 - Environments
- More abstraction
- Like and unlike CS3:
 - Language: Scheme
 - Good-bye UCWISE.
 - CS3 topics are covered in the first 3-4 weeks





CS61B: Data Structures

- Requires 61A (with a B- or higher)
- Language: Java
- Dynamic data structures:
 - Lists
 - Trees
 - Arrays
 - Strings
 - Hash tables
 - Queues
- Software engineering:
 - Design
 - Coding
 - Testing
 - Debugging
 - Analysis
 - There is a *lot* of programming in this course
- Run-time analysis
- Like less lecture, more hands-on? 61BL

CS61C: Machine Structures

- Requires: 61B
- Language: Mostly C
- Low-level programming:
 - What happens when you read data from a disk drive or hit a key on the keyboard?
- Machine architecture
- How operating systems actually work
- A little low-level programming (Languages: MIPS, Verilog)



CS70: Discrete Math/Prob. Theory

- Requires: Math 1B
- Proofs, logic problems, and algorithms
- Related to:
 - Cryptography
 - Networking efficiency
 - Search/sort algorithms
- No programming in this course, but...
- Weekly problem sets + a lot of proof and deduction problems
- Personal recommendation:
take it during/after 61B (some overlap)



The Upper Division

- ~4 Groups:
 - Hardware
 - Software
 - Theory
 - Application
- Focused on a single topic:
 - ✓ You can get really interested in a certain topic and get excited about it.
 - You can be forced into a class you don't really care about but have to do.
- These are usually where you do more of the “cool” projects (demo)



The Hardware Courses

- Enjoyed CS61C or EE40/42?
- **CS150: Digital Systems**
 - Low level work – circuits and logic elements
- **CS152: Computer Architecture**
 - Design of hardware systems
 - Hardware/software interface
 - How processors actually work





The Software Courses

- Enjoyed CS3, CS61A, or CS61B?
- **CS160: User Interface Design**
 - Design an interface in teams of 4-5 to a theoretical product
- **CS162: Operating Systems**
 - Add functionality in teams of 4-5 to NACHOS – an actual operating system
- **CS164: Programming Languages and Compilers**
 - The messy details on how your code actually gets turned into a runnable program
- **CS169: Software Engineering**
 - Design your own large-scale project in teams of 6-7.



The Theory Courses

- Enjoyed CS70?
- **CS170: Efficient Algorithms and Intractable Problems Design**
 - A more intense CS70
 - More on algorithms, computation problems
 - NP-complete problems
 - Problems that we think are unsolvable efficiently, but can't prove it
- **CS172: Computability and Complexity**
- **CS174: Combinatorics and Discrete Probability**




The Applications Courses

- Specific in their given areas
- **CS161: Security**
- **CS184: Computer Graphics**
- **CS186: Database Systems**
- **CS188: Artificial Intelligence**





More Courses

- CS9 self-paced: 1 unit, P/NP.
Learn another programming language:
 - Perl
 - Java
 - MatLab
 - C++
 - Unix
 - C
 - EE: analog versus digital building computers
 - There are many!
- 
- 
- 

The CS Major

- Interested in CS but want a little more freedom than the EECS major?
 - Double major is something non-technical.
 - Spend a semester abroad.
 - Too many other requirements in the College of Engineering.
- Luckily, there is a CS major in L&S
- ... and a CS minor for those who want even more freedom.



Tips for Success

- Don't fall behind.
 - CS is hard enough as it is.
 - Pay attention in lecture and/or read the book.
- Use your resources.
 - Engage your TAs.
 - Go to office hours.
 - Check out hkn.eecs.berkeley.edu for ratings.
- Form study groups.
 - Work gets done faster in groups.
 - More importantly, it gets done more correctly.
 - (Plus, it's more fun with friends.)

