An Overview of Computer Science and Engineering

Rafael Perez
Professor
Computer Science and Engineering

Ken Christensen
Assistant Professor
Computer Science and Engineering

Three tracks in the undergraduate CSE program

• Computer Science
  – Software oriented
• Computer Engineering
  – Hardware oriented
• Information Systems
  – Systems oriented

Hmmmmm.....

Three tracks in the undergraduate CSE program

• Key differences
  – CE has full engineering core
  – CS and CE have same math and physics
  – IS has reduced math and physics
  – IS has no hardware courses
  – CS and CE are accredited (by CSAB and ABET)
  – No accreditation standards exist for IS curricula anywhere
    » Accreditation standards are under work by IEEE and ACM

Which track is right for me?

• There really is not that much difference
  – In the end analysis all are a “computer science” degree
• However, a good way to choose is to look at possible career paths

Which track is right for me?

• A tree of career paths
  CSE curriculum tracks
  Development
  Systems, support
  Hardware
  Software
  IS
  NationsBank, etc.
  IBM, Intel, etc.
  Microsoft, Sun, etc.
Specialist versus generalist

- **Specialist** (few jobs, high salary)
  - Very competitive to obtain
  - Some require a M.S. or Ph.D.
  - Geographically constrained
  - High pressure and long hours
  - Very high pay
  - Very challenging
  - Professionally rewarding

- **Generalist** (many jobs, lower salary)
  - Easier to obtain
  - Not geographically constrained
  - Moderate pressure and hours
  - Lower pay
  - Possibly less challenging

Specialist versus generalist

Examples of a specialist job

- **CPU designer**
  - **Job**: design and develop processors for computers
  - **Education**: not a likely "entry level" position
  - **Location**: Silicon Valley, California
  - **Hours**: 80 hours per week (you choose working hours)
  - **Pay**: low six figures plus yearly bonuses
  - **Your work**: changes the world

- **Operating System designer**
  - **Job**: design and develop operating systems
  - **Education**: not a likely "entry level" position
  - **Location**: west coast
  - **Hours**: 80 hours per week (you choose working hours)
  - **Pay**: low six figures plus yearly bonuses
  - **Your work**: changes the world

Examples of a generalist job

- **System integrator**
  - **Job**: build computer systems for large companies
  - **Education**: some entry level opportunities
  - **Location**: any large city
  - **Hours**: 60 hours per week (lots of travel)
  - **Pay**: five figures plus yearly bonuses

- **End user support**
  - **Job**: keep users up and running and happy (!)
  - **Education**: entry-level opportunities
  - **Location**: any large city
  - **Hours**: 50 hours per week
  - **Pay**: five figures plus modest yearly bonuses

A day in the life of a developer

5:00am -- Get up
5:30 -- Commute (traffic is always bad in California)
6:30 -- First status meeting of the day (project is late, again, boss is yelling)
7:00 -- "Real work" at desk (debug code written late last night)
9:00 -- Cruise the web for a few minutes, but...
9:05 -- Panic call from the field, customer is going to another vendor
11:00 -- Off the phone (whew!) and some real work again
12:00pm -- Down to the vending machine (eat in the office, as always)
2:15 -- Find the bug in the program (a real stupid bug!), late to a meeting
3:30 -- Share rumors that project is to be cancelled
3:35 -- Share rumors that this year's bonus will be double last year's
4:30 -- Finished writing another module and off to the lab
7:30 -- End-of-day status meeting (boss is still yelling)
8:00 -- Start commute home
9:00 -- Do more work at home, pass-out at keyboard
A day in the life of a developer

Ohhhhh… I wish I could be a student again with all that spare time.

A day in the life of a support staff

Ohhhhh… I wish I was a student and didn’t have to deal with all these crazy people and unreliable systems.

Career planning

- Current trends
  - CPU design - Intel rules
  - Software - Microsoft rules
  - Systems - Compaq, HP, Sun, Dell, IBM
  - Support - Growing need
- Who does integration and support
  - Internal group?
  - Hired consultant?
  - Vendor who sells hardware and software?

Typical career path

- You really start here
  - Testing
  - Maintenance
  - Component development
  - Design
  - Design management (technical)
  - Management (corporate)
Some thoughts relative to a career

- **Upside (CS, CE, and IS):**
  - High pay
  - Lots of opportunities
  - Possibility to “start your own business”
  - Rapid change is exciting

- **Downside (CS, CE, and IS):**
  - “Full on” or “full off” - - - work very hard for very high pay
  - Rapid change is exhausting - - - little stability

- Make the right career decision
  - You will (likely) have to bet your career on a technology
  - Understand the marketplace and make a good bet

Admission to the CSE department

- Requirements for CSE tracks
  - Completion of gate courses as shown below

<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
<th>CE</th>
<th>CS</th>
<th>IS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Tools</td>
<td>EGN2210</td>
<td>Req</td>
<td>Req</td>
<td>Req</td>
</tr>
<tr>
<td>Program Design</td>
<td></td>
<td>Req</td>
<td>Req</td>
<td>Req</td>
</tr>
<tr>
<td>Computer Org</td>
<td></td>
<td>Req</td>
<td>Req</td>
<td>Req</td>
</tr>
<tr>
<td>Discrete Structure</td>
<td>EGT2100</td>
<td>Req</td>
<td>Req</td>
<td>Req</td>
</tr>
<tr>
<td>Eng Statics</td>
<td>EGN3310</td>
<td>Req</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thermodynamics</td>
<td>EGR23341</td>
<td>Req</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electrical Systems 1</td>
<td>EGR3373</td>
<td>Req</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering Statistics</td>
<td>EGN3443</td>
<td>Req</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Admission to the CSE department

- Fulfill all course requirements (obvious?)
  - GPA is 2.6 or better in the three department entry courses
  - Program Design, Computer Organization, and Discrete Structures

- Buy a PC (if you can)
  - Even an old “obsolete” (dirt cheap) PC is better than nothing
  - Start “experimenting” with things
  - Learn to use the Internet
  - Learn a little HTML (writing Web pages)
  - Learn a little programming in C (many, many good books)

- Develop a strong “work ethic” relative to studying
  - Poor “study skills” sink more students than anything else

Contact in Computer Science and Engineering:

Skip Brugger
Undergraduate Coordinator
ENB 342
skip@csee.usf.edu

Learn more about the CSE department