A LIGHTNING OVERVIEW OF THE PENN CIS UNDERGRADUATE CURRICULUM

March, 2010
Caveat

- These slides were assembled for the departmental advising fair in March, 2010. We’ve tried to check them for accuracy, but they probably still contain some errors, and they have not been updated since then to track changes in course scheduling and rostering.

- Check the departmental web pages for up-to-date information.
Core Courses
CIS 120  (Fall, Marcus / Spring, Weirich)

- Fast-paced introduction to Java programming for CIS majors
- Prerequisite: Some programming experience
  - E.g., a high-school programming course or CIS 110
CIS 121: DATA STRUCTURES AND INTRODUCTION TO ALGORITHMS IN JAVA

(Fall: Daniilidis, Spring: Tannen)

Prerequisites: 120 and 160!

Themes:
- become an even better programmer
  - 4-5 weekly prog. hwks, 1 multiweek project, Eclipse, JUnit
- worry about efficiency, analyze running time (Big-Oh)
  - 2 math-like hwks, math-like questions on exams
- learn the “data structures” set of techniques
  - Stacks, Queues, Lists, Trees, Heaps, Search Trees, Graphs, Hash Tables

Grade: two midterms, final, homework, lab participation

Spring 2010 Project: ???
CIS 160 (formerly CIS 260): Mathematical Foundations of Computer Science (Fall, Gallier/Spring, Nenkova)

- An introduction to basic proof techniques and mathematical reasoning as used in computer science.
- An introduction to basic mathematical concepts and techniques used in computer science.
- Mathematical reasoning and proof principles; logic
- Set theory
- Relations, Functions, Partial Functions
- Induction on the natural numbers
- Composition of relations and functions
- Recursion on the natural numbers
- Injections, Surjections, Bijections
- Inverses of functions and relations
- Equinumerosity, Cantor’s Theorem
- Pigeonhole principle, Schroeder-Bernstein Theorem
- Strings, Multisets
- An amazing surjection: Hilbert space-filling curve
- Some counting problems
- Binomial coefficients, the binomial formula
CIS 160 (continued)

- Partial orders, lattices
- Well-founded orderings and complete induction
- Equivalence relations and partitions. Closures.
- Trees, multiset ordering, string and tree embedding
- Graphs (directed and undirected; unlabeled and labeled)
- Graph homomorphisms
- Baby number theory; divisibility, modular arithmetic
- more …

- 5 homework assignments, 2 midterms, one final exam.
- Partial collaboration allowed.

- Ultimate goal: Learn how to attack problems effectively and how to construct rigorous arguments to validate proposed solutions. No nonsense approach, learn how to use counter-examples.
CIS240: Introduction to Computer Systems  
(Fall, Taylor)

• Prerequisite: CIS110

• Bottom-up, no-magic intro to systems
  • Data representation (1s and 0s)
  • Digital logic and hardware
  • Low-level programming and debugging
  • Compilers, operating systems, security

• Three programming projects
  • Operating system calls: in assembly
  • Game (Tetris, Maelstorm): in C
  • Binary utility: in C
CIS 261 - Discrete Probability, Stochastic Processes, and Statistical Inference  
(Mintz, Fall)

Prerequisite(s): CIS 160

The purpose of this course is to provide a 1 CU educational experience which tightly integrates the theory and applications of discrete probability, discrete stochastic processes, and discrete statistical inference in the study of computer science.
CIS 262: Theory of Computation
Instructor: Rajeev Alur, Fall

- Prerequisite: 160

The Science of Computing: Can we study computation abstractly (without worrying about specific programming language and computer platform)?

- There are inherent limits to what can be computed (the notion of Undecidability)
- Turing machines capture what can be computed
- Finite automata capture what can be computed with finite memory (and are useful for query languages)
- Tractable problems: What can be computed in polynomial-time (P vs NP question)
CIS 262: Theory of Computation

Topics:
- Finite automata and regular languages
- Context-free languages (CFL)
- Turing machines and recursive languages
- NP-completeness

Course emphasizes rigorous thinking and mathematical proofs
CIS 320: Introduction to Algorithms (Khanna, Spring)

Prerequisites: CIS 120, 121, 160, 262.

- Algorithm design paradigms
  - Central themes in designing efficient algorithms.
  - Commonly used algorithmic building blocks.

- Modeling problems
  - Identifying the core computational task.
  - Relating seemingly different problems.

- Limits of efficient computation
  - Recognizing intractability, and dealing with it.
Some of what you will learn ...

- How do you design a communication network?
- How do you find shortest paths in a map?
- How do you compress a large text file?
- How does diff3 work?
- How can you efficiently compute a good solution to the Traveling Salesman Problem (TSP)?
CIS 341: Compilers
(Zdancewic, Spring)

- Prerequisites:
  - CIS 121, 240

- Course topics:
  - Programming in Ocaml
  - Grammars, Lexing and Parsing, tools lex & yacc
  - Intermediate language representations
  - Syntax-directed translation, code generation
  - Implementing high-level features: loops, functions, closures, objects, exceptions
  - Optimization techniques: hoisting & code motion, CSE, constant propagation, strength reduction, register allocation,…
  - Memory management / garbage collection

- Heavily Project oriented: build a compiler
  - 80% projects / 20% exams

- Course projects will use Ocaml
  - Knowledge of Ocaml not a prerequisite
CIS371: Computer Organization and Design  
(Martin, Spring)

- Prerequisite: CIS240
- How hardware “really” works
  - Basic hardware tricks: caching, pipelining, speculation, parallelism
  - Performance, costs, and trade-offs
  - Experimental analysis
- Hardware prototyping project
  - Pipelined CPU using Verilog, FPGA
  - Ties in neatly with CIS240
CIS-380: Operating Systems (Blaze, Fall)

• Prerequisite: CIS-240 or EE-300
  – working knowledge of C programming essential

• What you’ll learn
  – How does software run, what’s an OS do?
  – Processes, memory, devices, storage, networks, security, distributed computing

• What you’ll do
  – 380 is course on OS principles (exams, short homeworks)
CIS400 / 401: Senior Project
(Fall/Spring, Lee)

- ‘nuff said...
CIS 455 – Internet and Web Systems
(Haeberlen, Spring)

- How do the systems of the Internet and the Web work – and how do they scale to millions of users?
- How do we build reliable, high-performance systems like EBAY, Google, Facebook, 2nd Life, …?
- How do we exchange and search for data in a distributed Web setting?
- How do we program large-scale distributed applications?

Prerequisites:
- Java skills, debugging skills, some familiarity with threads
- CIS 121 & CIS 380
CIS 455 Topics and Projects

- Architectures for distributing computation and data
  - client-server, hierarchical, P2P, cloud, grid, sensor, ...
- Programming models for distributed computing
  - servlet, message-passing, remote procedure calls, map/reduce, XQuery
- Fundamental distributed algorithms
  - clock sync, consensus, concurrency, transactions, ...
- XML and its role in the Web
  - XML for interchange and integration, XPath, XQuery, Web Services
- Web search
  - Information retrieval, keyword ranking, PageRank, Google
  “Programming in the large” – progressively build components to main project:
  - Web server; pub–sub system with crawler; P2P cache for YouTube results
  - Google clone – P2P crawler, index, search engine, PageRank, …
Electives and Advanced Courses
Networked Life will explore recent scientific efforts to explain social, economic and technological structures -- and the way these structures interact -- on many different scales, from the behavior of individuals or small groups to that of complex networks such as the Internet and the global economy.

This course covers computer science topics and other material that is mathematical, but all material will be presented in a way that is accessible to an educated audience with or without a strong technical background.

The course is open to all majors and all levels, and is taught accordingly. There will be ample opportunities for those of a quantitative bent to dig deeper into the topics we examine. The majority of the course is grounded in scientific and mathematical findings of the past two decades or less.
CIS-125: Technology and Public Policy  
(Smith/Blaze, Spring)

• How does technology affect our society? What’s the right way to deal with disruptive advances in technology? Can engineers be more powerful than politicians?
• We look at different technologies each week
  – Policy issues examined from engineering perspective
  – Engineering issues examined from policy perspective
• Non-engineers welcome
• Substantial writing and discussion
• Specific topics vary each semester, e.g.,
  – Privacy & surveillance
  – Nuclear weapons & arms control
    • And how to build your own bomb
  – Radio spectrum
  – Electronic voting
  – Nanotechnology
  – Biotechnology
  – Copyright and digital publishing
  – Regulating and funding science
CIS 140 - Introduction to Cognitive Science
(Ungar, Fall)

Cross listed with: Cogs001, Phil044, Ling105, Psych107

How do minds work? This course surveys a wide range of answers to this question from disciplines ranging from philosophy to neuroscience. The course devotes special attention to the use of simple computational and mathematical models. Topics include perception, action, thought, learning, memory and social interaction.

Lyle Ungar & David Brainard
This is a broad survey course, covering a wide range of models of mind. Particular attention is given to the visual system and to the role of Bayesian modeling.

CIS140 Topics

– What is intelligence?
– How do brains work?
– Learning and memory
– Voles in Love
– Perception
– Bayes Rule
– Concept Learning
– Development and Plasticity
– Attention and Executive Function
– Reinforcement Learning
– Judgment and Decision Making
– Mind, language and computation
– Emotion
– Why computers are autistic
– Metaphor
– Consciousness
CIS 19X: Mini Courses
(Spring)

- 1/2 credit, project-oriented, mostly skills and practice, not theory

- Prerequisites:
  - 120/121 (240 for C++ class)

- Each course covers a different topic:
  - C++ being taught Fall 2010
  - Unix skills
  - Python
  - C#
  - Haskell being taught Fall 2010
  - ?? (more as we can support)

- e.g. Web programming
What do the following topics all have in common?

- Computer Graphics
- Robotics
- Image Compression
- Control Theory
- Audio Processing

• They all revolve around numbers and numerical algorithms!
EAS 205 continued

- Course Facts
  - Prerequisites - Math 114 and CIS 110 or equivalent
  - Would count in Math column of CPG
EAS 205, Course Goals

- Explain how numerical methods are used to solve a variety of canonical problems such as:
  - Transform and lighting computation
    - Used in graphics, robotics, vision
  - Signal coding
    - Used in JPEG coding, Audio coding
  - PageRank
  - Face Recognition
  - Simulating Dynamical Systems

- Course will revolve around 5 or 6 projects and will cover the underlying mathematics and show how the theory is actually applied.
CIS/MKSE 212: Scalable + Cloud Computing

- What you will learn:
  - How large-scale web services such as Google or Facebook work behind the scenes
  - How to build extremely large-scale services (such as the file system Google uses)
  - How to write massively parallel programs and work with massive amounts of data
  - How to do social network analysis, e.g., to recommend new friends in Facebook
  - ... and much more!
  - Example projects: Mini-Flickr; YouTube-style movie recommendation

- About the course:
  - Prerequisites: CIS120, CIS160; co-requisite: CIS 121
  - Instructors: Zachary Ives and Andreas Haeberlen
  - Offered Fall 2010

© 2010 Andreas Haeberlen, Zachary G. Ives

University of Pennsylvania (March 19, 2010)
CIS 277: Introduction to Computer Graphics Techniques (DMD Required) (Badler, Spring)

- Programming the essential geometric concepts underlying 2D and 3D computer graphics.
- C++, OpenGL, QT, Python.
- Prerequisite CIS 120.
- Bridges the (large) programming gap between CIS120 (Java) and [significant] computer [graphics] programming projects [used in CIS 460].
- Stresses: code reuse, debugging, good program design, user interaction, challenge, “fun” assignments.
- Assignments build on one another.
Overview of C++, software library engineering and code debugging.
C++ pointers, classes, memory model, templates
Vectors, matrices, 2D transformation matrices (classes; manipulations)
Scene graph construction and traversal
QT, QT Designer, user interfaces, and event driven programming
Geometric algorithms: intersections, point-in-polygon, …
3D transformations and viewing transformations
Height fields; progressive mesh refinement (Google Earth)
Subdivision and fractal surfaces
Polygon mesh data structures
Bezier curves and surfaces
Shape deformation: global transformations and free-form deformations
2010 Assignments: Scene graph, 2D keyframe editor, “mini-Maya”
polygon mesh editor
EAS285 Teaching CS Concepts
(a service learning course)
Griffin, Burke

Prereq: CIS110
Summer, Spring
Learn CS while teaching middle/high school students

• READ about learning theory, computers & society
• LEARN technical skills
• CONSTRUCT new learning materials
• TEACH in the BootUp! after-school, summer camps
• Scratch animation
• Computational textiles
• BotWorld
• Unplugged computer science
• RESEARCH a topic of your choice

Fulfilling Penn’s Compact with Philadelphia
NSF-Funded
CIS330: Database Management Systems (Davidson, Spring)

• What’s the backbone of on-line reservation, registration, and retail systems (e.g. Amazon, eBay, Expedia,...)? Databases!

• Unlike other CS courses that focus on code, in this course the focus is on representing and managing data that lives beyond the lifetime of a program.
  – Reliable storage & recovery of 100s of GB
  – Querying/updating interface
  – Fast, optimized retrieval
  – Support for many concurrent users
  – APIs for web applications

• Course involves SQL, XQuery and other homeworks and a project (done in groups) involving relational db design, SQL, XML, Java programming

• Offered in the fall, prerequisites are CIS121, 160

• See http://www.seas.upenn.edu/~cse330/
CIS 334: Randomized Algorithms (Khanna, Fall)

Prerequisite: CIS 320.

- A powerful algorithmic resource, like space and time.
- Simple algorithms that match/exceed the performance of complex deterministic algorithms.
- Algorithms and protocols for problems that we are unable to solve deterministically.
Some of what you will learn …

- How do you check if two large documents are identical by examining a small number of bits?
- How do you check the output of a program without actually doing the computation yourself?
- How does Google compute Pagerank?

Learn about some neat research areas: approximation algorithms, data streaming, interactive proofs, online algorithms, random walks, secure computation.
Large-scale software engineering:
- Large versus small systems
- Project Scheduling: Planning vs. Reality
- Internal Communication & Documentation
- Programming in teams/groups
- Planning, Coding and Testing
- Tools – version control, HLL, toolkits
- Manuals, User Interfaces (look and feel)


Grading:
- 20% Project Plan
- 20% Mid-term exam
- 20% Final Exam
- 40% Project Evaluation

Project (currently Web 2.0 focused):
- Run as a software startup
- 2-4 person groups
- System admin and test groups
- Application groups focus on adding new services, such as: ATMs near Penn, Food trucks, Facebook event planner, Career services events, etc.
- Weekly “all-hands” meetings with 5-slide presentations by each group

System for Penn Active Maps (SPAM)
http://spam.seas.upenn.edu
CIS 368 (néé 399-06) (Nimeroff, Spring): User Interfaces and the Web

- Fundamentals of Human-Computer Interaction (HCI)
  - Theory
  - Design
  - Implementation
  - Experimentation
  - Evaluation

- In the context of current web interaction mechanisms, technologies and applications

- Leverage open source technologies to develop sound, compelling user interfaces, especially in social networking environments
CIS 390, MEAM 420/520 Robotics (Shi, Fall 2010)

Main topics:
1) kinematics of robot
2) computer vision for estimation and control
3) planning motions for mobile robots
4) learning for improving performance

Robotics is a fast moving field. Making a robot think and act is no longer science fiction. Honda's Asimo Humanoid robot, Sony Aibo robotic dogs and DARPA Grand Challenge robotic cars are exciting examples. It is an interdisciplinary topic covering concepts from computer science, electrical engineering and mechanical engineering. One goal of this course is to expose students to the connections among CS, EE and ME through a sequence of hands-on projects which involving designing, building and controlling a mobile robot. Students will be encouraged to form interdisciplinary teams across different majors.

Project demo: http://www.youtube.com/watch?v=LIWKXva4d10
Prerequisites: CIS 121

Focus: Artificial Intelligence as Building Rational Agents

- Learn methods & programming techniques to build rational agents: Programs that can achieve a set of goals or maximize a payoff, given the available information
What You’ll Learn & Build

0. Introduction & Python Programming
1. Search Strategies, Constraint Satisfaction, Adversarial Games
   • Build a bot that competes against other game bots to play a simple board game (Kingdoms)
   • Build a solver for any Sudoku problem
3. Knowledge Representation and Logic
   • Build an agent that plays Hunt the Wumpus using propositional logic theorem proving
5. Machine Learning: Perceptrons, Naïve Bayes, etc.
   • Build a spam filter using Naïve Bayes machine learning
CIS 398 - Quantum Computer and Information Science
(Mintz, Spring)

Prerequisite(s): CIS 160, CIS 262, and Math 240.

The purpose of this course is to introduce undergraduate students in computer science and engineering to quantum computers and quantum information science. This course is meant primarily for juniors and seniors in CIS. No prior knowledge of quantum mechanics is assumed.
CIS 460/560: Computer Graphics
(DMD Required: Badler, Fall)

- Conceptual and programming overview of computer graphics focusing on 3D modeling and realistic rendering.

- Geometric transformations, geometric algorithms, software systems (OpenGL), 3D object models (surface and volume), visible surface algorithms, image synthesis, shading and mapping, ray tracing, radiosity, volume rendering, global illumination, photon mapping, and anti-aliasing.

- Prerequisite: CIS 277 (C++) or equivalent programming experience (1 year+).

- Programming assignments: scene graph, 3D modeler, ray-tracing, volume (“cloud”) rendering.
CIS 462/562: Computer Animation
(DMD Required: Safonova, Fall)

Description:
Solid technical foundation for developing, animating and controlling articulated systems used in interactive computer games, virtual reality simulations and high-end animation applications.

Topics include:
• Principals of Animation and Keyframing
• Motion capture
• Procedural animation
• Facial animation
• Computer simulation
• Crowd animation
• Intelligent agents
• Forward and inverse kinematics
• Dynamic systems and control
• And others…
CIS 462: Computer Animation

Last Year Projects

Simple Particle System of shooting rockets and generating sparks
Motion Capture, Forward Kinematics
Inverse Kinematics, Motion Blending

Also:
• A curve editor using Bezier curve, Hermite curve and B-spline
• An agent/crowd simulator

Prerequisites:
CIS 120
Familiarity with linear algebra (or EAS 205)
CIS 277 preferred

New Vicon Motion Capture Lab
This course is primarily meant for undergraduate and graduate students from Linguistics, Psychology, Philosophy, and Biology.

CIS undergraduates will be allowed only if they agree to do a substantial term project involving computational linguistics.

-- This requirement is there because a lot of the mathematical work in this course is already familiar to the CIS undergraduates.

-- This course is not a substitute for CIS 160, 261, or 2XX (Introduction to Computational Linguistics).

http://www.seas.upenn.edu/~cse477
CIS 477-Ling 549: Mathematical Techniques in Linguistics

• Basic techniques in mathematical linguistics focusing on formal grammars and machines, as applied to several aspects linguistics such as morphology, syntax, semantics, and even discourse
• These topics will be discussed in a very basic and introductory manner, focusing more on detailed linguistic examples, directly related to computation of linguistic structures

How many interpretations for

I ate sushi with tuna
I ate sushi with tuna

I bet you didn’t think of this interpretation before!
CIS 497 (néé EAS499): DMD Senior Design (DMD Required)

• The goal of this course is to provide an opportunity to define, design, and execute an individualized project.

• Evaluation is based on ability to:
  – select an interesting topic
  – Communicate objectives in writing and presentations
  – accurately gauge what resources are required to complete chosen task
  – Execute the plan.
500-Level Courses
CIS 500, Software Foundations  
(Pierce, Spring)

- A rigorous introduction to the theory of programming languages
  - functional programming
  - constructive logic and the Coq proof assistant
  - reasoning about programs (program equivalences, Hoare logic)
  - type systems and type safety

- Prerequisites: 121, 160, 262; substantial mathematical maturity
CIS520: Machine Learning  (Taskar, Fall)

How does your email know what is spam?  
or your camera focus on faces?  
or your phone recognize what you said?  
or helicopter-robot fly upside down?

How can you make machines learn to work better?

Course content
- Learning algorithms, theory and applications
- Some topics: regression, support vector machines, boosting, neural nets, generalization bounds, EM, k-means, PCA, dimensionality reduction, HMMs, graphical models, reinforcement learning, ...

Fine print (prerequisites)
- Multivariate calculus, linear algebra, elementary probability
- Programming experience (C, Java, or Matlab)
Grand Challenge: For computers to use NL as effectively as people
- Reading and writing text: Summarizing, Extraction into Databases
- Interactive Dialogue: Natural, effective access to computer systems; Informal Speech Input and Output
- Translation: Input and Output in Multiple Languages

Topics:
- I – Intro & Word–Based Methods: Introduction to Python, Statistics; Statistical Part of Speech Tagging; Speech Recognition
- II – Parsing: Introduction to Syntactic Analysis; Statistical Parsing
- III – Extracting Meaning: Automatic Word Sense Disambiguation, Sense Role Labeling, Discourse Structure
- IV – Putting the Pieces Together: Machine Translation, Summarization

Prerequisites: An intro to AI, Machine Learning, Cognitive Science or Natural Language syntax or permission of the instructor
CIS 534: Multicore Programming  
(not being taught ’10-’11)

• Prerequisites: CIS371, CIS380
  • C++ experience recommended

• Dominant paradigm for next 5-10 years: multi-core chips
  • CPUs: few cores today, dozens tomorrow
  • GPUs: already highly parallel

• How do we program them?
  • Active area of research

• Course topics
  • Identifying & extracting parallelism, parallel hardware, data-, task-, thread-level parallelism, synchronization, performance tuning, hands on projects
CIS 540: Principles of Embedded Computation
Instructor: Rajeev Alur
Spring 2011

- Prerequisite: mathematical maturity (ESE 210 or CIS 262)
- Fairly new course
- Principles underlying design and analysis of computing elements that interact with physical environment
  - Reactive: Interaction of a component with its environment via inputs/outputs
  - Communication, Concurrency, Real-time response
  - Sensors, actuators, computers all together
  - Canonical examples: Cruise control software
CIS 540: Principles of Embedded Computation

- Theory + Assignments in MATLAB
- Topics
  - Model-based Design
    - State machines, Dynamical systems, Hybrid systems
  - Models of Interaction
    - Asynchronous, Dataflow, Time-triggered
  - Specification and Verification
    - Requirements, Analysis techniques, Model checking
  - Modeling Performance
    - Probabilistic models, Steady-state analysis, Quantitative properties (e.g. power usage)
Prerequisites:
- Java (CIS 120 / 121)
- C or C++ helpful (CIS 240)
  (but not necessary - you can pick up what you need to know)
- Some computer networks experience (helpful but not required)

Note: Undergraduates *should* take 551
(Old 331 is now merged with this class)

Projects:
- Buffer overflow attack
- Network intrusion detection (filter TCP traffic)
- Simplified DES implementation
- Simulated ATM: authentication, authorization, key exchange, signatures

2 Midterms and a Final
CIS 551 Course Topics

- Software Security / Malicious Code
  - Buffer overflows, viruses, worms, protection mechanisms

- System Security
  - Hacker behavior, intrusion & anomaly detection, hacker and admin tools

- Networks & Infrastructure
  - TCP/IP, Denial of Service, IPSEC, TLS/SSL

- Internet Security
  - Viruses, worms, spam, web security (XSS), phishing

- Basic Cryptography
  - Shared Key Crypto (AES/DES), Public Key Crypto (RSA)

- Crypto Software & Applications
  - Cryptographic libraries, authentication, digital signatures

- Covert Channels
CIS 553: Networked Systems
(Loo, Fall)

- Instructor: Boon Thau Loo (boonloo@cis.upenn.edu)
- Course website: http://www.cis.upenn.edu/~boonloo/cis553-fa07/
- How are networks designed, tested, and built? How do routers work? How can we build overlay networks over the Internet?
- Topics covered: Internet architecture, routers, transport protocols (TCP), network security, p2p networks, wireless networks, overlay networks, network testbeds (PlanetLab), network simulation/simulation tools
- Course pre-requisites: CSE 121 or equivalent.
- Students design and build a large networked system, layer-by-layer on a 24-node cluster:
  - Router implementation: Link state and Distance Vector protocols
  - Chord Distributed Hash Table over their routing code
  - P2P applications (e.g. keyword search, publish/subscribe) over Chord
CIS 553 Projects

- Undergraduates are welcomed!
  - 7 undergrads last year, out of 34 students. All did very well.
  - 3 Penn undergrads and 2 masters students working on independent studies with Prof. Loo this semester.
    - One has submitted an ACM SIGCOMM workshop paper!
    - Another is doing summer internship at a startup company collaborating with Prof. Loo.

- Next year’s highlights:
  - TCP: students build a reliable transport protocol over their routing protocol implementation
  - Network Simulator (NS-3) open-source development
    - http://www.nsnam.org/
CIS 563: Physically Based Animation

When: Spring 2011 (Safonova)

Description:
- Learn state-of-the-art techniques used to produce highly realistic special effects for:
  - movies
  - video games
  - surgical simulation systems

These are examples of what you will learn to simulate!
CIS 563: Physically Based Animation

Learn Physically Based Modeling techniques for the animation of:

  - fluids and gases
  - explosions
  - virtual characters
  - rigid and deformable solids
  - cloth
  - fire and other systems

Prerequisites:
Good knowledge of object-oriented programming
Familiarity with linear algebra and physics
Intended for undergraduate students!

Website for this class:
http://www.cis.upenn.edu/~alla/courses/cis563/
(contains a full syllabus and other information)
CIS 564 (Lane, Summer/Fall)

Game Design and Development

- Course Overview
  - Provides students with a solid theoretical understanding of the core creative principles, concepts, and game play structures underlying most game designs (i.e. “Theory of Games”)

- Course also examines game development from an engineering point of view, including:
  - game play mechanics,
  - game engine software and hardware architectures,
  - user interfaces,
  - design documents, play-testing and production methods
CIS 565: GPU Programming and Architecture (not taught again till 2011)

1. Understanding of the GPU as a graphics pipeline
2. Understanding of the GPU as a high performance massively parallel multi-core compute device
3. Understanding of various GPU architectures
4. Programming in: CG, CUDA and OpenCL
5. Exposure to many core graphics effects performed on GPUs
6. Exposure to many core parallel algorithms performed on GPUs that enable dramatic increases in computing performance
CIS 568 (Lane, Summer/Fall)
Game Design Practicum

- Prerequisites: CIS462/562, CIS277 or CIS460/560, Co-requisite: CIS564
- Experience designing and developing 3D computer games.
- Work in teams of three or four, brainstorm an original game concept, write a formal game design document then develop a fully functional prototype consisting of a playable level of the game.
- Create original art and animation assets for the game.
- Develop technical features including: a novel game mechanic and/or user interaction model, game physics (i.e., particle systems and rigid body dynamics), character animation, game AI (i.e., movement control, path planning, decision making, etc.), sound effects and background music, 2D graphical user interface (GUI) design and optional multiplayer networking capabilities.
- Use C++, Python, Lua, industry middleware and game engines.
- Execute design iteration through user feedback and play-testing, and assess what makes a game fun to play.
CIS 581: Computer Vision and Computational Photography
(Shi, Fall)

Main Topics:
1) Image features: edge detection
2) Image morphing
3) Image matching and stitching
4) Image recognition

This course is intended to provide you a hands-on experience with interesting things to do on images/videos. The world is becoming image-centric. Camera are now found everywhere, in our cell phones, automobiles, even in medical surgery tools. Computer vision technology has lead to latest innovations in areas such as Hollywood movie production, medical diagnosis, biometrics, and digital library. This course is suited for students with all Engineering background, who has the basic knowledge of linear algebra and programming, and a lot of imagination.

Grading Policy: 3 homeworks/projects 60%, Midterm 20%, Final Project 20%.
Computer Engineering Major
New: Computer Engineering

- Computer Engineering (CE)
  - New major coming online Fall 2009
  - Can switch in as a sophomore (easy from CIS)
  - Hands on, lab oriented, not for the faint of heart

- CPG
  - CIS 110, 120, 121, 160, 240, 371, 380
  - ESE 200, 201, 205, 215 (electronics)
  - Year-long sophomore lab: iPhone
  - Junior-fall lab: medical device
  - Junior-spring lab: massively parallel application
CIS441/CIS541: Embedded Software for Life-Critical Applications (Fall 2010)

- Prerequisite: CIS240; ESE350 recommended
- The course is
  - The course is to study principles, methods, and techniques for building *Cyber-Physical Systems* that are safety critical.
- The goal is
  - to give students greater design and implementation experience in life-critical embedded software development, and
  - to teach them how to model, design, optimize, verify, implement, and validate safety critical systems in a principled manner.
- Topics covered include
  - Cyber physical systems, distributed real-time systems, real-time programming, assurance cases, modeling and verification, testing and validation, software architecture...
- The team project is to design and implement a life-critical system such as a pacemaker, and consists of six tasks:
  1. modeling
  2. verification
  3. implementation
  4. validation
  5. demo
  6. a written report based on
ESE250: Digital Audio Basics

• **Informally:** Understanding iPod/iPhone -- overview of the technology behind cell phones and MP3 players
• ½ CU course, moved to Spring semester
• Pre-req.: programming course (CIS110 ok)
• DeHon and Koditschek
• Part of new Computer Engineering Major
• Lead-in to revised ESE350 (spring) where will build a complete digital-audio platform

http://www.seas.upenn.edu/~ese250/
Coda
Computing & Educational Technology Services

Windows labs: Moore 100b, c, Towne m62, m70
Unix labs: Moore 100a, 207
Remote Windows access: virtual lab
Remote Unix access: eniac.seas.upenn.edu
Email, web service, networking, course software
FAQs: [http://www.seas.upenn.edu/cets/answers](http://www.seas.upenn.edu/cets/answers)
Talk to instructors about homework problems
Office and free printing location at 169 Moore
Email [cets@seas.upenn.edu](mailto:cets@seas.upenn.edu) for lab problems and suggestions
The Dining Philosophers: Undergrad CS Club

• We aim to create a CS community at Penn – curriculum talks, research presentations, invited speakers, programming contests, even just hanging out and talking about anything!
• We’re looking for new members! Leadership opportunities are available to anyone
• Meetings: Weds. Nights, 9pm, Rodin lobby
• General info session w/ free food in the next few weeks – watch your email
• Coming soon: a special breakfast with a technology industry VIP for members only