A LIGHTNING OVERVIEW OF THE PENN CIS UNDERGRADUATE CURRICULUM

March, 2016
Caveat

- Check the departmental web pages for up-to-date information.
Core Courses
CSCI Core Curriculum

- CIS 110: Introduction to Computer Programming
- CIS 120: Programming Languages & Techniques I
- CIS 121: Programming Languages & Techniques II

Programming

- CIS 160: Mathematical Foundations for Computer Science
- CIS 262: Automata, Computability, and Complexity
- CIS 320: Introduction to Algorithms

Theory

- CIS 240: Introduction to Computer Architecture
- CIS 371: Computer Organization and Design
- CIS 380: Computer Operating Systems

Systems

- CIS 400/401: Senior Design Project

(Exact requirements vary with the degree program)
CIS 120  
(Fall, Zdancewic / Spring, Pierce & Marcus)

• Fast-paced introduction to programming for CIS majors
• Prerequisite: Some programming experience
  – e.g., a high-school programming course or CIS 110
• Topics:
  - program design, testing
  - lists, trees, recursion
  - abstraction, abstract datatypes: sets, maps, containers
  - heap-structured data, references, aliasing
  - object-oriented design
• Approach:
  - ½ OCaml, ½ Java
  - 10 projects, including: Phylogenetic trees, GUI implementation, Image Processing, Game
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-O)
  - 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
CIS 160: Mathematical Foundations of Computer Science
(Fall, Gandhi/Spring, Tannen)

- 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 …

- Many 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.
CIS 262: Theory of Computation
Instructor: Roth, Fall/Gallier, Spring

- 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
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)?
CIS240: Introduction to Computer Systems
(Staff – Fall, Farmer – Spring)

• 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
CIS371: Computer Organization and Design
(Brown, 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 (Loo, 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, projects)
Small groups work together on an engineering problem of their choice.
  - Showcase creativity and expertise

It is what you make of it... 'nuff said...
Project Electives

Note: only the following courses may be used to satisfy the "Project Elective" requirement on the CPG. Other courses, even if they have a significant coding project, are not approved.
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 the “Quaker OAT” compiler
  72% projects / 28% exams
CIS 350 Software Design & Engineering  
(Murphy, Spring)

- You know how to write a "program".
- But how do you create a software "product" as part of a team, with customers that have expectations of functionality and quality?

- Topics include:
  - software development lifecycle models
  - software architecture and design
  - Android application development
  - software testing
  - software maintenance
CIS450/550: Database Management Systems
(Davidson, Fall / Staff, 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
- Prerequisites are CIS121, 160
- ***FORMERLY CIS 330***
CIS 455/555 – Internet and Web Systems
(Ives, Fall / 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, ...
CIS 460/560: Introduction to Computer Graphics Techniques (DMD Required) (Brown,Fall)

- Programming the essential geometric concepts underlying 2D and 3D computer graphics. Prerequisite CIS 120.
- C++, OpenGL, and QT/QT Designer.
- Extends programming beyond CIS120 (Java) into [significant] computer [graphics] programming projects
  - used in CIS 461
- Stresses: code reuse, debugging, good program design, graphical user interfaces, challenge, “fun” visual assignments.
- Assignments build on one another: Geometric algorithms, scene graphs, keyframe editor, 3D camera viewer, polygon mesh editor, subdivision surfaces, deformations.
- ***FORMERLY CIS 277*** will be taught in Fall from now on
CIS 553: Networked Systems  
(Smith, Spring)

• 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
ESE350 Introduction to Embedded Systems

Ever wanted to build some a flying quadrotor, an arcade game or learn how to network wireless sensors, controllers and actuators?

ESE350 is where hardware marries software and the world of Embedded Systems begins.

A great course for CIS/CIT juniors and seniors. No hardware skills necessary just knowledge of basic C programming

Offered every Spring.
Check out http://www.seas.upenn.edu/~ese350/

The Internet of Things – when sensors, controllers and actuators talk to each other
CIS Undergraduate Electives
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
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

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.
CIS 19X: Mini Courses
(Various - Fall/Spring)

• 1/2 credit, technology-oriented, mostly skills and practice, not theory
• Prerequisites: 120/121
• Not available for preregistration
• Each course covers a different topic:
  – 190 - C++ (also requires CIS 240)
  – 191 - Linux/Unix skills
  – 192 - Python
  – 194 - Haskell
  – 195 - iPhone App Development
  – 196 - Ruby on Rails Web Development
  – 197 - Javascript
  – 198 - Rust
CIS 331: Intro. to Networks & Security
(Heninger, Spring)

• Basic concepts:
  – threat models, the security mindset
• Applied cryptography:
  – public and private key encryption, digital signatures and authentication, hash functions, secure channels
• Networks and network security:
  – IP, TCP, routing, network protocols, web architecture, web attacks, firewalls, intrusion detection
• Software security:
  – defensive programming, memory protection, sandboxing, virtual machines, buffer overflows, malware
• Broader issues:
  – privacy, security and the law, digital rights management, denial of service, ethics, etc.
• Prerequisites: CIS 160, CIS 240
CIS 390  
(Daniliidis, Fall)

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

• ***FORMERLY CIS 391***
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
2. Knowledge Representation and Logic
   - Build an agent that plays Hunt the Wumpus using propositional logic theorem proving
3. Machine Learning: Perceptrons, Naïve Bayes, etc.
   - Build a spam filter using Naïve Bayes machine learning
CIS 399 Open-Source Software Development
(Murphy, Fall)

- Students work on open-source projects for academic credit under supervision of faculty member and industry mentor
- Mentor provides guidance on process, feedback on code, etc.
- Weekly reading assignments cover cultural, technical, and legal aspects of open-source software
- Students must apply to be admitted to course; an email about this will go out soon
CIS 419/519: Intro. to Machine Learning (Eaton, Fall)  www.seas.upenn.edu/~cis519

Introduces machine learning algorithms and concepts, emphasizing the practical application of ML to real problems

Scene Labeling via Deep Learning  Reinforcement Learning for Mario  www.youtube.com/watch?v=4cgWya-wjgY  Grouping Patients by Genomic Sequences

**Format:** Individual written/implementation assignments, exams, and a team project applying ML to open problems

**Prerequisites:** CIS 121
Statistics or linear algebra would be helpful, but not required
CIS441/CIS541: Embedded Software for Life-Critical Applications (Lee, Fall)

- 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 assurance case.
CIS 461/561: Computer Graphics (DMD Required: Badler, Mally, Spring)

- Conceptual and programming overview of computer graphics focusing realistic image synthesis.
- Physics-based rendering, shading, ray tracing, radiosity, volume rendering, global illumination, photon mapping, anti-aliasing, geometric transformations, geometric algorithms, 3D object models (surface and volume), algorithm efficiency, and visible surface algorithms.
- Prerequisite: CIS 277/CIS 460 (C++, fundamental graphics concepts) or equivalent programming and graphics experience (1 year+).
- Programming assignments: Ray tracer, physics-based path tracer, renderer optimization.
CIS 462/562: Computer Animation
(DMD Required: Lane, 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 462/562: Computer Animation

Student 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 277/460/560 preferred

New Vicon Motion Capture Lab
CIS 497: DMD Senior Design (DMD Required) (Badler: Fall/Spring)

• 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.
NETS

Networked and Social Systems
NETS112: NETWORKED LIFE
(Kearns, Fall)

• 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.
How do social networks make recommendations?

How has Google rewritten the rules of advertising, using concepts from auctions and game theory?

How does a search engine work?

How does the Internet work?

If you’ve wondered about any of these... then NETS 150 is the course for you!

- It’s about networks – social networks, information networks, commerce networks, communications networks

- Prerequisites: CIS 110; co-requisites: CIS 120; 160 rec.
NETS 212: Scalable + Cloud Computing (Haeberlen – Fall)

- **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 videos on YouTube
  - ... and much more!
  - Projects: Mini-Facebook, LiveJournal visualizer, MapReduce on EC2

- **About the course:**
  - Prerequisites: CIS120, CIS160; co-requisite: CIS 121
NETS 412: Algorithmic Game Theory (Roth - Spring)

- Topics:
  - Game Theory, Game Dynamics
    - Linear Programming, zero sum games, Nash equilibria
  - Mechanism Design
    - Market equilibria, auctions, ...

- Prerequisites:
  - CIS 320 (may be taken concurrently)
CIS-125: Technology and Public Policy
(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
• Counts as TBS
CIS-125 (continued)

• 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 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.
EAS 205, Applications of Numerical Methods  
(Taylor / Fall)

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, 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.

• Prerequisites - Math 114 and CIS 110 or equivalent
• Counts as a mathematics course
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.
• Counts as a Natural Science
Advanced Elective Courses
CIS 500, Software Foundations

(Pierce, Fall)

• 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
CIS520: Machine Learning
(Ungar, 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)
"Evidence of the existence of secret compartments" behind the tomb of Tutankhamen.

Egyptian Antiquities Minister Mamdouh Eldamaty said that radar surveys indicate by "90 percent" to the possible existence of two chambers did not Tkchwa behind Tutankhamun's tomb. The searches conducted by the archaeologists in order to find the tomb of Queen Nefertiti, wife of King Akhenaten and Tutankhamun's father, to the possible existence of "organic material" inside the empty spaces behind the walls in the king's tomb.

Learn how Google Translate works! This hands-on project based class will teach you about machine translation, which draws from machine learning, linguistics, probability and statistics.

Undergraduates are welcome! See more at http://mt-class.org/penn/
CIS 530: Computational Linguistics  
(Nenkova, Spring)

**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
Prerequisite: mathematical maturity (ESE 210 or CIS 262)

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)

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 552: Advanced Programming
(Weirich, Spring)

- Goals:
  - to take good programmers and turn them into excellent ones
  - to introduce a range of modern software engineering practices

- Content: Mindbending projects in Haskell, an advanced functional programming language

- Prerequisites: Four courses involving significant programming and a discrete mathematics or modern algebra course

- Audience: Juniors and Seniors, Grad students

http://www.seas.upenn.edu/~sweirich/cis552/
CIS 556  Cryptography
(Heninger, Fall)

- **Symmetric cryptography** Block ciphers, stream ciphers, hash functions
- **Public-key cryptography** number theory, public key encryption, digital signatures
- **Cryptographic security** network security protocols, side-channel attacks
- **Magical crypto tricks** secret sharing, commitments, zero-knowledge proofs
- **Research topics** privacy-enhancing technologies, lattices, etc.

**Prerequisites:** Mathematical maturity, comfort with both proofs and programming. CIS 320+262 is probably sufficient, Math 371 is helpful.
CIS 557: Programming for the Web
(Swapneel Sheth, Fall)

• Ruby and Ruby on Rails (Formerly, CIT 597)
• “Culture” of Web Programming
  – Software Engineering for the Web (Agile and XP, BDD)
  – Software as a Service, REST, and MVC
  – Software Testing with BDD and TDD
  – Using Web Services and APIs
  – Deployment and Monitoring

• First half of the semester – individual assignments;
Second half of the semester – team project
• Prerequisites: CIS 121
• http://cis.upenn.edu/~cis557/
CIS 565: GPU Programming and Architecture (Cozzi, Fall)

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, Fall/Spring)  
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 573 Software Engineering (Murphy, Fall)

• Focus: “What is good code?”
• Considers quantitative aspects of software quality (internal and external)
• Emphasis on refactoring and testing
• Project in which students continue development of existing Android app
• Pre-req: CIS 350 or significant industry experience
Grading Policy: Homeworks 50%, Midterm 20%, Final 30%
Homeworks on paper and using Matlab.
Midterm is on paper.
Final is a project including paper calculations and Matlab.

Zisserman and Hartley, Multiple View Geometry
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%.
Tell your friends

Accessible courses for non-engineers
CIS 105 Computational data exploration  
(Fall, Bhusnurmath)

• Intro to data science for non majors. Learn programming via data science applications.
• Will not count for engineering credit
• No prerequisite
• Topics
  – Data gathering
  – Exploratory data analysis
  – Data visualization
  – Simulations
  – Linear regression

We will cover the programming concepts required in order to achieve these tasks
CIS 106 Visualizing the Past/Peopling the Past

• Fall 2016 (cross-listed with ANTH 258, ANTH 620)
  – Co-taught by Badler (CIS) and Erikson (Anthropology)

• Topics:
  – Fundamentals of programming and data analysis (Python)
  – 3D modeling (Audodesk Maya) and animation (Unity3D)
  – In-depth exposure to ancient culture, such as the Inca, through scholarly resources and popular media

• Approaches fundamental issues in Anthropology and Computer Science from both inside each field as well as from outside using the perspective of the other
• Will not count for engineering credit
Coda
2 linux labs, 4 Windows labs
Free Microsoft software via MSDNAA for enrolled students (everything except Office)
Pennkey-protected web pages
Opt-in CGI (PHP, perl, wiki, etc.)
Backups, snapshots
Remote access to Virtual Windows lab
Mount unix filesystem on Windows lab computers
Lifetime email forwarding via alumni system
Calling Penn Engineering Students...

Are You Interested In Studying Abroad?

- Fulfill your major requirements while exploring a different part of the world.
- Increase your independence and confidence as you immerse yourself in a new environment.
- Develop your intercultural abilities, a valued skill in a globalized world.
- It's a great way to enhance your academic experience.

To get started:

Contact Caryn Stivelman
Associate Director of Undergraduate Affairs
chstiv@seas.upenn.edu