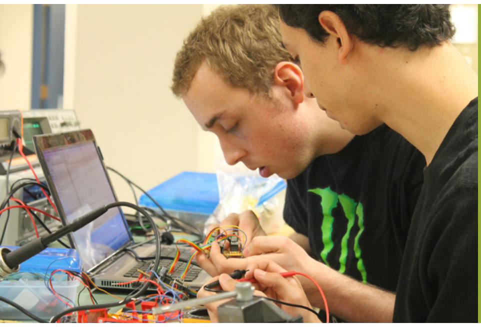
Northwestern | McCORMICK SCHOOL OF ENGINEERING

Undergraduate Study Manual



Electrical Engineering & Computer Science Department

2017-2018

TABLE OF CONTENTS

1 Welcome from the Chair

My name is Alan Sahakian, Chair of EECS, and I am pleased to inform you about the opportunities available to you as an undergraduate in our department.

To major in Electrical Engineering (EE), Computer Engineering (CE), or Computer Science (CS) is to choose an excellent and lucrative career path, with many options available to you after graduation. Our graduates immediately go to work at such illustrious companies as Google, Microsoft, Pixar, Motorola, Mobility, Tesla Motors, Apple, Ford, General Motors, IBM, Intel, Yahoo, Sun, HP, and Northrup Grumann. Some have started their own companies. Many go on to graduate school, and a degree from Northwestern assists in placement in top graduate programs for PhD and professional degrees.

Currently our undergraduate majors number about 250. Many work in our research labs alongside professors and graduate students doing real research and publishing papers. Our students are involved in Design Competition, developing competing robots, building and racing NU's solar car, creating games and numerous other exciting projects. Economics, optoelectronics, journalism and technology, networking and communications, linguistics, graphics, networks, theory, distributed computing -- and more are found within EECS. Take a look at our research groups list, which is always evolving.

Degree options are wide-ranging. Besides degrees in CS, EE and CE, we offer also an option to get your Masters degree simultaneously with your BS degree, or to choose a course of study in Computer Science within the Weinberg School of Arts & Sciences, as an alternative to the McCormick School of Engineering.

Non-majors can also learn about computers and programming by making use of our NU Programming curriculum, an invaluable resource.

Our popular mentoring program links up undecided freshmen with upperclassmen at a meal at a local restaurant, paid for by the department. Read more about it on our page for First-Years.

I invite you to explore our website and learn more about us. If you have questions, or would like specific information about one or more of our programs, please phone us at 847-491-3451 or email *mentors@eecs.northwestern.edu*.

We look forward to hearing from you!

Sincerely,

Alan V. Sahakian,
Professor and Chair

1.1 Welcome from the Associate Chair of the EECS Undergraduate Program

As associate chair of the undergraduate program for the Department of Electrical Engineering and Computer Science, it is a pleasure to welcome you to the department. The EECS department is the largest department in the McCormick School with well over 50 full-time faculty, who have numerous distinctions in both their teaching and research. The size of the department is due in part to the enormous breadth of EECS, which includes artificial intelligence, computer and communications systems, human-computer interaction, nanoelectronics, and photonics. Technologies and systems based on these areas are pervasive in modern society and provide essential tools for improving the lives of people around the world.

EECS offers undergraduate degrees in computer science, computer engineering and electrical engineering. These degrees can lead to a wide range of career paths including careers in traditional EECS industries such as software or hardware companies, graduate studies in EECS or related fields, and careers in other fields such as law, medicine, journalism, consulting, and finance. A number of exciting multidisciplinary studies can be pursued within EECS. For example, in collaboration with the Medill school, students can participate in a multidisciplinary project course on innovation in Journalism and Technology. As another example, Electrical engineering provides a biomedical track to facilitate a biomedical or premedical focus. Other ties include strong connections to Cognitive Science, Economics, Mechanical Engineering, the Learning Sciences and Physics. We also offer the possibility of a Computer Science degree through Weinberg school of Arts and Sciences, enabling one to pursue a liberal arts background while obtaining a CS degree. In addition to our majors, EECS also offer a minor in computer science for both McCormick and Weinberg students, providing a way for students to develop a strong competence in computer science while major in another area.

The degree programs in EECS are continually being examined and improved. This work is led by curriculum committees for each major, which are currently chaired by Prof. Robby Findler for Computer Science, Prof. Russ Joseph for Computer Engineering, and myself for Electrical Engineering. Recent changes include completely re-vamping EECS 203, the introductory course in computer engineering as well as changing the format of EECS 101, an introductory course in computer science.

Again, welcome to EECS. I look forward to meeting each of you and following your progress during your studies.

Sincerely,

Randall Berry, Ph.D.
Professor of EECS
Associate Chair of the EECS Undergraduate Program

2. Electrical Engineering Curriculum

Electrical Engineering involves the development and application of electronic and optical technologies for generating, communicating, and processing information. Our EE curriculum includes courses in electronic circuits, solid-state electronics, electromagnetics, optics, lasers, controls, digital signal processing, communications and networks.

Individual engineers may work in one or more of a large number of functional capacities in support of a total engineering effort. For example, engineers may choose system design and specification, component design, research and development, university teaching and research, consulting, production and quality control, sales, cost analysis, or management.

Students may specialize in any of a number of areas, including:

- Circuits and Electronics
- Solid State Engineering
- Electromagnetics and Photonics
- Systems including Digital Signal Processing, Communications and Control

The electrical engineering program is accredited by the Engineering Accreditation Commission of ABET, http://www.abet.org.

2.1 Mission Statement for our Undergraduate Program in EE

The Electrical Engineering (EE) program involves the design and analysis of electronic devices and circuits, photonics, electromagnetics, and analog and digital systems, including control, communication, and information systems. It encompasses several broad areas and a core of fundamental knowledge, as well as many subfields of specialization.

The goal of the EE undergraduate program is to educate electrical engineers in the fundamentals and applications of electrical engineering via a curriculum that allows sufficient flexibility to encompass graduates directly entering the work force and graduates pursuing graduate education. Graduates of the program should have a solid foundation in the theory underlying the field as it's practiced, and be able to communicate effectively both in oral and written forms. A distinguishing feature of our program is the fact that the EE department at Northwestern is relatively small, which allows small class sizes and close interaction between students and faculty.

Electrical Engineering Undergraduate Mission Statement
To educate undergraduates in the basic principles and modern practices of the field of electrical engineering and train our students to think independently, to master the systematic approach to problem solving, and to have a keen awareness of the role of engineering in a modern technological society.

The broad objectives we expect graduates to obtain from our program are:

- Career Preparation: Graduates will apply their electrical engineering skills to a variety of challenges in industry, academia or in the pursuit of other fields.

- Professionalism and leadership: Graduates will attain careers in which they become leaders in their chosen fields, work in multi-disciplinary teams, make decisions that are socially responsible, and communicate effectively.

- Intellectual curiosity: Graduates will continuously learn new concepts, identify new directions, and adapt in response to the needs of a rapidly changing world.

To prepare our graduates to achieve these objectives, we intend for students of the Electrical Engineering program to graduate with the following knowledge and skills:

1. Knowledge of continuous and discrete math
2. Knowledge of core Electrical Engineering topics
3. An ability to use modern engineering techniques for analysis and design
4. An ability to apply knowledge of math, science and engineering to the analysis of Electrical Engineering problems
5. Knowledge of probability and statistics
6. An ability to design and conduct scientific and engineering experiments, as well as to analyze and interpret data
7. An ability to design systems which include hardware and/or software components
8. An ability to identify, formulate and solve novel Electrical Engineering problems
9. An ability to function in multidisciplinary teams
10. An understanding of ethical and professional responsibility
11. An ability to convey technical material through oral presentation and interaction with an audience
12. An ability to convey technical material through formal written papers and reports
13. A broad education and knowledge of contemporary issues
14. A recognition of the need for, and an ability to engage in, life-long learning
15. The ability to get a good job or admission to a top graduate school.

2.2 Background on Electrical Engineering Curriculum

In the year 2000, in response to feedback from our students, employers, and alumni, we designed a new electrical engineering curriculum.

The common themes in our EE curriculum are:

1) Reduced total number of required courses to allow more flexibility.
2) Freshman/Sophomore level courses which provide broad overviews of the fields of electrical engineering and computer engineering.
3) Several fundamentals courses to provide in-depth introductions to various sub-fields of electrical engineering; these courses would also form the pre-requisites for all subsequent advanced courses in those sub-fields.
4) Exciting hands-on labs and computer labs to complement all our lecture classes.
5) An exciting curriculum and courses relevant to current applications of electrical engineering.

6) Requiring students to do team-based design projects and encouraging students to do undergraduate research.

The Electrical Engineering curriculum is compatible with a premedical program of study. If you are interested in this option, please discuss it with your academic adviser.

We offer two courses that are suitable for freshmen and sophomores and are required of both EE and CE majors and provide a one quarter overview of the fields of computer engineering and electrical engineering along with exciting labs involving the design of a robot and a CD player.

- *EECS 202: Introduction to Electrical Engineering*
- *EECS 203: Introduction to Computer Engineering*

We also offer five fundamentals courses:

- *EECS 221: Fundamentals of Circuits*
- *EECS 222: Fundamentals of Signals and Systems*
- *EECS 223: Fundamentals of Solid-State Engineering*
- *EECS 224: Fundamentals of Electromagnetics and Photonics*
- *EECS 225: Fundamentals of Electronics*

These five courses are required of all EE students and provide fundamental knowledge in each field of electrical engineering. Subsequently, students will be able to take the rest of the technical electives from a wide range of choices in each field.

In addition all EE students are required to take one of the capstone design class projects and encouraged to take two 399 independent research units.

- *EECS 347-1: Microprocessor Systems Design Projects*
- *EECS 347-2: Microprocessor Systems Projects*
- *EECS 392: VLSI Systems Design Projects*
- *EECS 398: Electrical Engineering Design*
- *EECS 399: Project*

An overview of the electrical engineering curricular concept is illustrated in Figure 1.

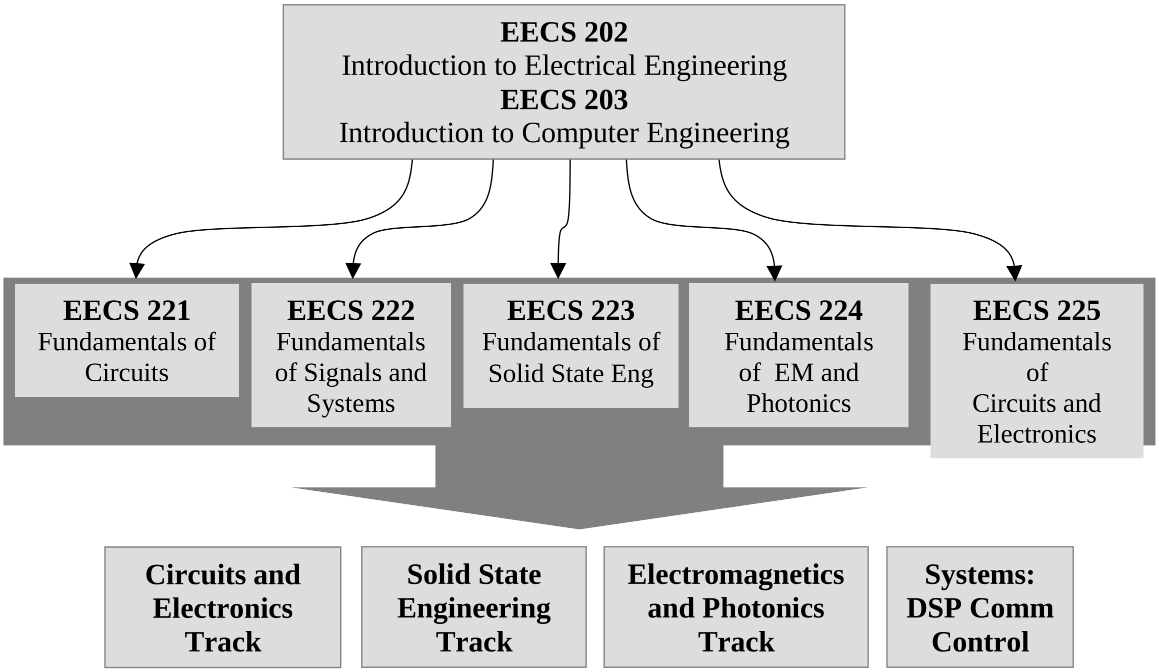

EECS 202
Introduction to Electrical Engineering
EECS 203
Introduction to Computer Engineering
EECS 221
Fundamentals of Circuits
EECS 222
Fundamentals of Signals and Systems
EECS 223
Fundamentals of Solid State Eng
EECS 224
Fundamentals of EM and Photonics
EECS 225
Fundamentals of Circuits and Electronics
Circuits and Electronics Track
Solid State Engineering Track
Electromagnetics and Photonics Track
Systems: DSP Comm Control

Figure 1. An Overview of our EE Curriculum.

2.3. Details of the Electrical Engineering Curriculum

Total Requirements - forty-eight courses

Engineering Analysis - four courses

GEN_ENG 205-1: Engineering Analysis 1 - Computational Methods and Linear Algebra
GEN_ENG 205-2: Engineering Analysis 2 - Linear Algebra and Mechanics
GEN_ENG 205-3: Engineering Analysis 3 - Dynamic System Modeling
GEN_ENG 205-4: Engineering Analysis 4 - Differential Equations

Engineering Design and Communications – three courses

DSGN 106- 1, 2 (.5 units each)
ENGLISH 106- 1, 2 (.5 units each)
1 unit from COMM ST 102 or PERF ST 103 or PERF ST 203

Mathematics - four courses

Math 220,224,230: Calculus I, II, III
Math 234: Multiple Integration and Vector Calculus

Basic Sciences - four courses

Physics 135-2,3: General Physics
Two additional science courses (The courses selected must be consistent with the list of approved Basic Sciences for the McCormick school.)

Basic Engineering - five courses

Electrical Engineering:

EECS 202: Introduction to Electrical Engineering

Computer Architecture and Numerical Methods:

EECS 203: Introduction to Computer Engineering

Programming

EECS 230: Programming for Engineers

or

EECS 211: Fundamentals of Programming (C++)

Probability, Statistics, and Quality Control:

EECS 302: Probabilistic Systems and Random Signals

One course chosen from the following four categories:
(The courses selected must be consistent with the list of approved basic engineering courses for the McCormick school.)

Thermodynamics
Fluids and Solids
Systems Engineering and Analysis
Materials Science

Unrestricted Electives - five courses

Social Science - Humanities Requirement - seven courses

Electrical Engineering Departmental Program - five courses

EECS 221: Fundamentals of Circuits
EECS 222: Fundamentals of Signals and Systems
EECS 223: Fundamentals of Solid State Engineering
EECS 224: Fundamentals of Electromagnetics and Photonics
EECS 225: Fundamentals of Electronics

Technical Electives - ten courses

Technical Electives can be used to tailor a program to a particular area of specialization. What follows are recommended courses for specialization in each of the EE tracks. At least six of the ten technical electives must be chosen from the following list of courses. Two additional courses must be 300-level Technical elective courses from the EECS department, or Biol 210-1,2,3 or Chem 210-1,2,3, or from the list of courses below. The remaining two courses can be 300-level Technical courses from science, mathematics, computer science or engineering courses and may include the following courses.

Circuits and Electronics

EECS 303: Advanced Digital Logic Design
EECS 353: Digital Microelectronics
EECS 391: Introduction to VLSI Design
EECS 393/493: Design and Analysis of High-Speed Integrated Circuits
EECS 355: ASIC and FPGA Design
EECS 346: Microprocessor System Design
EECS 347-2: Microprocessor Systems Projects

Solid State Engineering

EECS 250: Physical Electronics and Devices
EECS 381: Electronic Properties of Materials
EECS 384: Solid State Electronic Devices
EECS 385: Optoelectronics
EECS 388: Nanotechnology
ME 381: Introduction to Microelectromechanical Systems

Electromagnetics and Photonics

EECS 308: Advanced Electromagnetics and Photonics
EECS 379: Lasers and Coherent Optics
EECS 382: Photonic Information Processing
EECS 383: Fiber-Optic Communications
EECS 386: Computational Electromagnetics and Photonics

Systems

- **Digital Signal Processing**

EECS 332: Digital Image Analysis
EECS 359: Digital Signal Processing
EECS 363: Digital Filters

- **Communications Systems**

EECS 307: Communications Systems
EECS 333: Introduction to Communications Networks
EECS 378: Digital Communications
EECS 380: Wireless Communications

- **Control**

EECS 360 – Introduction to Feedback Systems
or
ME 391: Fundamentals of Control Systems
EECS 374: Introduction to Digital Control
EECS 390: Introduction to Robotics
ME 333: Introduction to Mechatronics

- **Biomedical Systems**

BME 325: Medical Imaging
BME 383: Cardiovascular Instrumentation
BME 327: Magnetic Resource Imaging
BME 333: Modern Optical Microscopy and Imaging
BME 317: Biochemical Sensors

Electrical Design Requirement – one course

To satisfy the Department's EE design capstone course requirements, students must elect to take at least one course from the following menu:

EECS 398: Electrical Engineering Design
EECS 392: VLSI Design Projects (391 is prerequisite)
EECS 347-1: Microprocessor System Projects (346 is prerequisite)
EECS 399: Project (where the 399 is structured as a design project)*

*Students must file a form, included in this handbook, for this 399 to be counted as the design requirement.

At most two units of EECS 399 will be allowed as Technical electives and one as the design requirement in the Electrical Engineering curriculum. Additional units of EECS 399 may be taken, but will be counted as unrestricted electives.

Upon completion of the design project, students are required to prepare a report using the template provided at http://www.eecs.northwestern.edu/images/docs/forms/capstonedesign.pdf

ELECTRICAL ENGINEERING P/N POLICY STATEMENT:

Among the 16 departmental courses, the P/N option may only be used within the ten technical electives. In addition, students may only have two P or D grades in the 16 departmental courses.

REQUIREMENT FOR GRADUATION In addition to McCormick graduation requirements, a student must receive a C– or better in EECS 202 and EECS 203 in order to continue in the EE program.

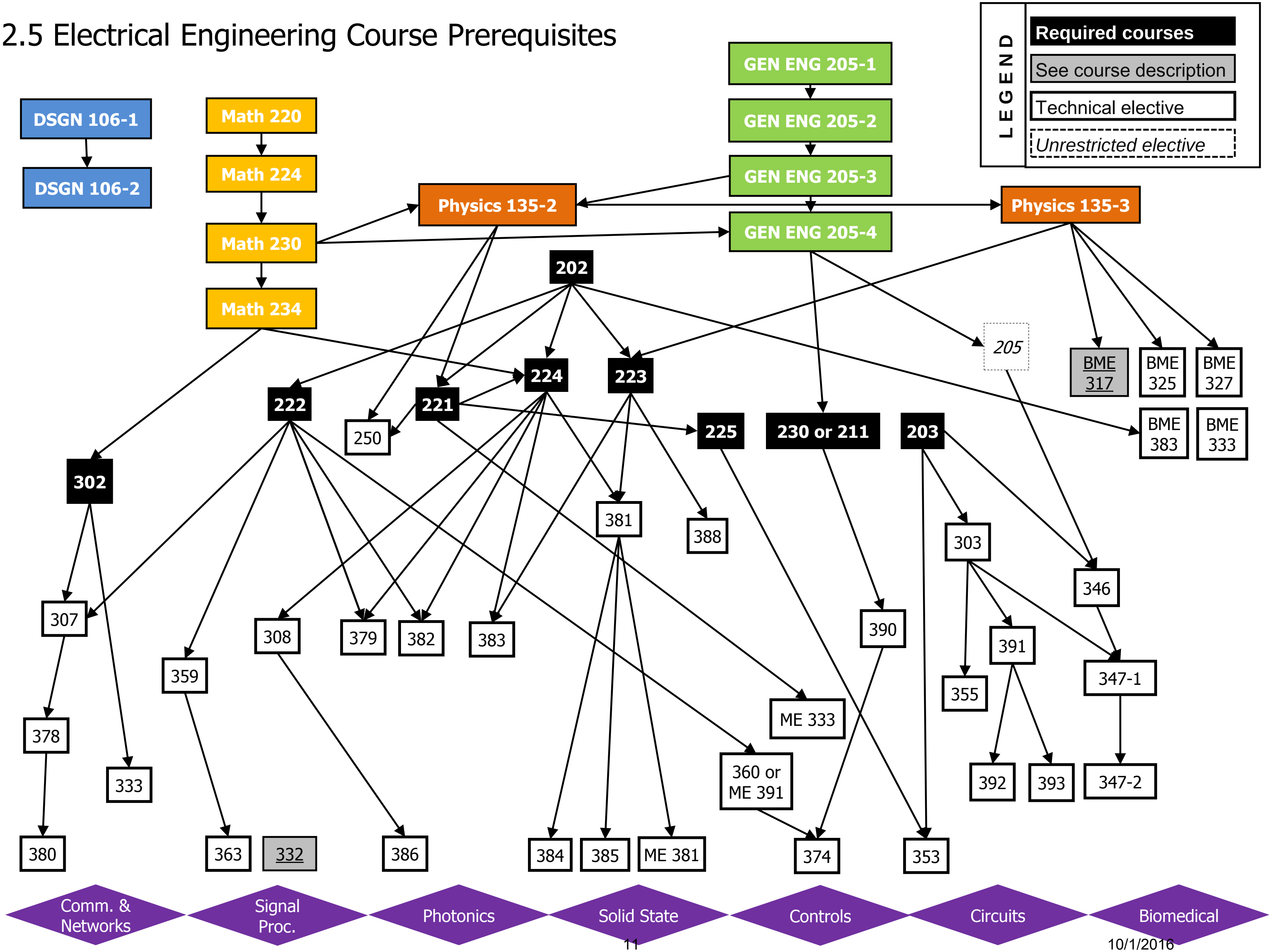
2.4 Preferred Schedule for Electrical Engineering

2011-2012

	FALL	WINTER	SPRING
Freshman	Math 220 Chem-101 GenEng 205-1 COMM ST 102 or 103	Math 224 Chem-102 GenEng 205-2 DSGN- 106-1	Math 230 EECS 203 (F,W) GenEng 205-3 DSGN - 106-2
Sophomore	EECS 202 (W,S) Physics 135-2 GenEng - 205-4 SS/Hum	EECS 221 (S) Physics 135-3 Math 234 SS/Hum	EECS 223 (F) EECS 230 (W) EECS 302 (F) SS/Hum
Junior Non Co-op	EECS 222 (W) EECS 224 (S) SS/Hum	EECS 225 (F) Tech Elective Tech Elective SS/Hum	Tech Elective Tech Elective Unres Elective SS/Hum
Senior Non Co-op	Tech Elective Basic Eng. Unres Elective SS/Hum	Tech Elective EE Design Tech Elective Unres Elective	Tech Elective Tech Elective Unres Elective Unres Elective
Junior Co-op	Tech Elective Tech Elective Tech Elective SS/Hum	Tech Elective Tech Elective Tech Elective SS/Hum	Work
Pre-Senior Co-op	Tech Elective Basic Eng Tech Elective SS/Hum	Work	Tech Elective Tech Elective Unres Elective SS/Hum
Senior Co-op	Work	Tech Elective Tech Elective Unres Elective Unres Elective	Tech Elective Tech Elective EE Design Unres Elective

*** Note that 221 and 223 must be taken before 225**

2.5 Electrical Engineering Course Prerequisites



LEGEND

- Required courses
- See course description
- Technical elective
- *Unrestricted elective*

Name

2.6 ELECTRICAL ENGINEERING STUDY PLAN

2010-2011

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
General Engineering, Mathematics, Basic Science																		
G_E 205-1																		
G_E 205-2																		
G_E 205-3																		
G_E 205-4																		
DSGN 106-1																		
DSGN 106-2																		
Math 220																		
Math 224																		
Math 230																		
Math 234																		
Phy 135-2																		
Phy 135-3																		
Basic Engineering																		
EECS 202																		
EECS 203																		
EECS 230																		
EECS 302																		
Communications & Social Science/Humanities																		
G_C102 or 103																		

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Departmental Program																		
EECS 221																		
EECS 222																		
EECS 223																		
EECS 224																		
EECS 225																		
Capstone Design																		
Technical Electives																		
Unrestricted																		

2.7 Conformance of the Electrical Engineering Curriculum with ABET Guidelines

The following table indicates how the Electrical Engineering curriculum conforms to the guidelines established by ABET Engineering Accreditation Commission.

Year & Quarter	Course (Department, Number, Title)	*Curricular Area (quarter course units)*			
		Math & Basic Sciences	Engineering Topics Check if Contains Design (√)	General Education	Other
1, Fall	Math 220, Calculus	1			
1, Fall	Basic Science Elective	1			
1, Fall	GEN_ENG-205-1, Computational Methods and Linear Algebra	0.8	0.2		
1, Fall	General Communication 102 or 103			1	
1, Winter	Math 224, Calculus II	1			
1, Winter	Basic Science Elective 2	1			
1, Winter	GEN_ENG_205-2, Linear Algebra and Mechanics	0.5	0.5		
1,Winter	DSGN 106-1, Engineering Design and Communications I		0.5 √	0.5	
1, Spring	Math 230, Vectors and Partial Derivatives	1			
1, Spring	GEN_ENG-205-3, Dynamic System Modeling	0.8	0.2		
1, Spring	DSGN 106-2, Engineering Design and Communications II		0.5 √	0.5	
1, Spring	EECS 203, Intro to Computer Engineering		1 √		
2, Fall	EECS 202, Intro to Electrical Engineering		1		
2, Fall	Physics 135-2, Electricity and Magnetism	1			
2, Fall	GEN_ENG-205-4, Differential Equations	0.9	0.1		
2, Fall	Hum/Social Science Elective 1			1	
2, Winter	EECS 221, Fundamentals of Circuits	0.25	0.75		
2, Winter	Math 234 – Calculus and Analytical Geometry	1			
2,Winter	Physics 135-3, Waves	1			
2, Winter	Hum/Social Science Elective 2			1	
2, Spring	EECS 223, Fund of solid-state	0.25	0.75		
2, Spring	EECS 230, Programming for Eng or EECS 211, Fundamentals of Programming (C++)		1 √		
2, Spring	EECS 302, Probability	0.5	0.5		
2, Spring	Hum/Social Science Elective 3			1	
3, Fall	EECS 222, Fundamentals of Signals and Systems	0.25	0.75		

3, Fall	EECS 224, Fundamentals of Electromagnetics and Photonics	0.25	0.75		
3, Fall	Technical Elective 1		1 √ *		
3, Fall	Social Science Elective 4			1	
3, Winter	EECS 225, Fundamentals of Electronics		1 √		
3, Winter	Technical Elective 2		1 √ *		
3, Winter	Technical Elective 3		1 √ *		
3, Winter	Social Science Elective 5			1	
3, Spring	Technical Elective 4		1√ *		
3, Spring	Technical Elective 5		1√ *		
3, Spring	Unrestricted Elective 1			1	
3, Spring	Social Science Elective 6			1	
4, Fall	Technical Elective 6		1√ *		
4, Fall	Basic Engineering elective		1		
4, Fall	Social Science Elective 7			1	
4, Fall	Unrestricted Elective 2			1	
4, Winter	Technical Elective 7		1√ *		
4, Winter	Technical Elective 8		1√ *		
4, Winter	Capstone Design		1√		
4, Winter	Unrestricted Elective 3			1	
4, Spring	Technical Elective 9		1√ *		
4, Spring	Technical Elective 10		1√ *		
4, Spring	Unrestricted Elective 4			1	
4, Spring	Unrestricted Elective 5			1	
TOTALS-ABET BASIC-LEVEL REQUIREMENTS		12.5	21.5	14	
OVERALL TOTAL QUARTER UNITS FOR THE DEGREE : 48					
PERCENT OF TOTAL		26%	44.8%	29.2%	
Total must satisfy either credit hours or percentage	Minimum Quarter Units	12 Units	18 units		
	Minimum Percentage	25%	37.5 %		

*Many but not all technical electives contain engineering design.

2.8 EECS 399 Design Requirement Form

Electrical Engineering and Computer Science Department
McCormick School of Engineering
Northwestern University
Evanston, Il 60208

Request for Approval to use EECS 399 as design credit in the EE major

This form is to be used to recognize the use of EECS 399 as meeting the departmental design requirement for Electrical Engineering majors and thus should only be filled out for Electrical Engineering Majors. This form should not be confused with a permission form to take EECS 399.

It should be completed by the instructor and given to the student at the time of giving the student the permission number for EECS 399. All information on the form must be filled out. The student should register on line for EECS 399 and submit this completed form to the Academic Services Office, L269, in order to have the 399 posted as design credit.

Please print clearly:

Students Name: ____

Student's Empl Id (7 digit ID #): __________________________

Student’s Major: _________________________________

Section Number:_________________________________

Quarter and Year taken:_____________________________________

By signing this form, the instructor certifies that this 399 meets the ABET definition of a major design experience in Electrical Engineering, including engineering standards and realistic constraints.

Instructor's Name: (printed)____

Instructor's Signature: ____

Date: _______________________________________

3. Computer Engineering Curriculum

The Computer Engineering program teaches the design of complex digital systems, from transistors to software systems. It deals with digital circuit and system design, computer architecture, robotics, microprocessors, software systems, and embedded systems. The interrelationships between and appropriate roles of hardware and software are emphasized. Our Computer Engineering curriculum involves courses in digital logic, electronic circuits, computer architecture, robotics, VLSI design, VLSI CAD, software development, operating systems, microprocessor-based systems, and parallel computing.

Students who are interested in pursuing a curriculum in computing that emphasizes understanding of computer hardware and the hardware/software interface should sign up for a B.S., M.S. or Ph.D. degree in Computer Engineering in the EECS Department.

Computer engineers have broad professional employment opportunities including design and management responsibilities, working with microchips and computers, application-specific hardware-software systems, computer-aided design (CAD) tools for digital systems, aerospace systems, defense systems, and networked systems.

Our Computer Engineering curriculum has strong lab-based learning emphasis and culminates in three design-projects-based courses. Interested undergraduates can get involved earlier in significant project or research work. Our teaching laboratories have recently been upgraded with the latest computer workstations, computer-controlled instruments and new experiments in newly renovated labs.

The Computer Engineering curriculum allows students to focus on a particular area of specialization. The areas include

- High-Performance Computing
- VLSI and Computer Aided Design
- Embedded Systems
- Software

The computer engineering program is accredited by the Engineering Accreditation Commission of ABET, http://www.abet.org.

3.1. Mission of our Undergraduate Program in Computer Engineering

The Computer Engineering (CE) program involves the design and engineering of computers including hardware and software design. It is a carefully chosen synthesis of computer engineering, computer science, and electrical engineering courses to train students how do design complex digital systems, from transistors to software. Computer engineering is a broad area involving many possible areas of specialization. These include Computer Architecture, VLSI Systems, Computer-Aided Design, Software Design, Robotics, Computer Vision, and Embedded Systems.

Computer Engineering Mission Statement

To educate undergraduates in the basic principles and modern practices of the field of computer engineering, and train our students to think independently, to master the systematic approach to problem solving, and to have a keen awareness of the role of engineering in a modern technological society.

The broad objectives we expect graduates to obtain from our program are:

1. Career Preparation: Graduates will apply their electrical engineering skills to a variety of challenges in industry, academia or in the pursuit of other fields.

2. Professionalism and leadership: Graduates will attain careers in which they become leaders in their chosen fields, work in multi-disciplinary teams, make decisions that are socially responsible, and communicate effectively.

3. Intellectual curiosity: Graduates will continuously learn new concepts, identify new directions, and adapt in response to the needs of a rapidly changing world.

To prepare our graduates to achieve these objectives, we intend for students of the Computer Engineering program to graduate with the following knowledge and skills:

1. Knowledge of continuous and discrete math
2. Knowledge of core Computer Engineering topics
3. An ability to use modern engineering techniques for analysis and design
4. An ability to apply knowledge of math, science and engineering to the analysis of Computer Engineering problems
5. Knowledge of probability and statistics
6. An ability to design and conduct scientific and engineering experiments, as well as to analyze and interpret data
7. An ability to design systems which include hardware and/or software components
8. An ability to identify, formulate and solve novel Computer Engineering problems
9. An ability to function in multidisciplinary teams
10. An understanding of ethical and professional responsibility
11. An ability to convey technical material through oral presentation and interaction with an audience
12. An ability to convey technical material through formal written papers and reports
13. A broad education and knowledge of contemporary issues
14. A recognition of the need for, and an ability to engage in, life-long learning
15. The ability to get a good job or admission to a top graduate school.

3.2 Background on Computer Engineering Curriculum

Our curriculum is continuously revised based on feedback from our constituents, e.g., our advisory board, industrial affiliates, and students. The common themes in the Computer Engineering curriculum follow:

1) A moderate number of required courses to allow flexibility in plans of study.

2) Freshman/Sophomore level courses that provide broad overviews of the fields of electrical engineering and computer engineering.

3) Exciting hands-on labs and computer labs to complement lectures.

4) A curriculum and courses relevant to current applications of computer engineering.

5) Requiring team-based design projects and encouraging undergraduate research.

The Computer Engineering curriculum is compatible with a premedical program of study. If you are interested in this option, please discuss it with your academic adviser.

We offer two courses that are suitable for freshmen and sophomores and are required of both EE and Computer Engineering majors. These courses provide one-quarter overviews of the fields of electrical engineering and computer engineering along with exciting labs involving the design of a robot and a CD player.

- *EECS 202: Introduction to Electrical Engineering*
- *EECS 203: Introduction to Computer Engineering*

These six courses have been identified as essential to Computer Engineers and are required.

- *EECS 111: Fundamentals of Computer Programming I*
- *EECS 211: Fundamentals of Computer Programming II*
- *EECS 205: Fundamentals of Computer Systems Software*
- *EECS 221: Fundamentals of Circuits*
- *EECS 303: Advanced Digital Logic Design*
- *EECS 361: Computer Architecture*

Computer Engineers are also required to take two of the following relevant courses:

- *EECS 213: Introduction to Computer Systems*
- *EECS 222: Fundamentals of Signals and Systems*
- *EECS 223: Fundamentals of Solid-State Engineering*
- *EECS 224: Fundamentals of Electromagnetics and Photonics*
- *EECS 225: Fundamentals of Electronics*

These five courses provide fundamental knowledge in each field of electrical engineering. Subsequently, students will be able to take some of the Technical Electives from a wide range of choices in each field within the field of EE.

In addition all Computer Engineering students are required to take one of the three capstone design classes listed below:

- EECS 347-1, 347-2: Microprocessor Systems Design Projects
- EECS 362: Computer Architecture Project
- EECS 392: VLSI Systems Design ProjectsComputer Engineering students are encouraged to take two 399 independent research units. They should also consider taking a graduate-level course in an area of interest as a technical elective.

An overview of the computer engineering curricular concept is illustrated in Figure 2.

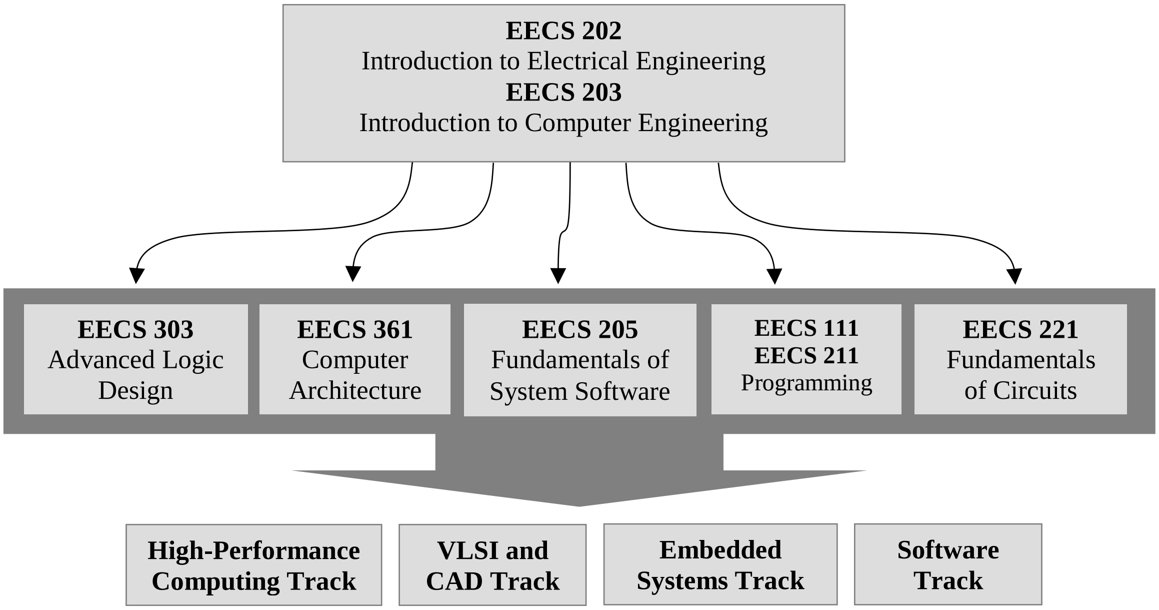


Figure 2. An Overview of our Computer Engineering Curriculum.

3.3. Details of Computer Engineering Curriculum

Total Requirements – forty-eight courses

Engineering Analysis – four courses

GEN_ENG-205 1: Engineering Analysis 1 – Computational Methods and Linear Algebra
GEN_ENG 205-2: Engineering Analysis 2 – Linear Algebra and Mechanics
GEN_ENG 205-3: Engineering Analysis 3 – Dynamic System Modeling
GEN_ENG 205-4: Engineering Analysis 4 – Differential Equations

Engineering Design and Communications – three courses

DSGN 106- 1, 2 (.5 units each)
ENGLISH 106- 1, 2 (.5 units each)
1 unit from COMM ST 102 or PERF ST 103 or PERF ST 203

Mathematics – four courses

Math 220,224,230: Calculus I, II, III
Math 234: Multiple Integration and Vector Calculus

Basic Sciences – four courses

Physics 135-2,3: General Physics

Two additional science courses. (The courses selected must be consistent with the list of approved Basic Sciences for McCormick.)

Basic Engineering – five courses

Electrical Engineering:

EECS 202: Introduction to Electrical Engineering

Computer Architecture and Numerical Methods:

EECS 203: Introduction to Computer Engineering

Programming

EECS 211: Fundamentals of Computer Programming II

Probability, Statistics, and Quality Control:

EECS 302: Probabilistic Systems and Random Signals

One course chosen from the following four categories:

(The courses selected must be consistent with the list of approved basic engineering courses for the McCormick school.)

Thermodynamics
Fluids and Solids
Systems Engineering and Analysis
Materials Science

Unrestricted Electives – **five courses**

Social Science - Humanities Requirement – **seven courses**

Computer Engineering Departmental Program – **five courses**

EECS 111: Fundamentals of Computer Programming I
EECS 205: Fundamentals of Computer System Software
EECS 303: Advanced Logic Design
EECS 221: Fundamentals of Circuits
EECS 361: Computer Architecture

Technical Electives – ten courses

Students must take at least five courses from the following four areas and two courses from the Fundamental EECS courses listed below. The remaining three Technical Electives can be any 300-level course from science, mathematics, computer science or engineering and may include the following courses. Biol 210-1,2,3 and Chem 210-1,2,3 may be used as Technical Electives. Furthermore, 400-level courses can be used to fulfill Technical Elective requirements with a petition and advisor consent. Some suggested examples are included in the lists below. For all courses listed below students should check the yearly course offerings.

Technical Electives can be used to tailor a program to a particular area of specialization. In the following, suggested courses for each area are listed. It is not required to follow a specific area in its entirety, they are suggestions to better structure one's major. Also, this is not an exhaustive list and other related 300- and 400-level courses can be taken based on advisor approval.

At most two units of 399 will be allowed as Technical Electives in the Computer Engineering curriculum. Additional units of EECS 399 may be taken, but will be counted as an unrestricted elective.

Area 1. High-Performance Computing

EECS 328: Numerical Methods for Engineers
EECS 333: Introduction to Communication Networks
EECS 339: Introduction to Database Systems
EECS 350: Introduction to Computer Security
EECS 354 - Network Penetration and Security
EECS 358: Introduction to Parallel Computing
EECS 362: Computer Architecture Projects
EECS 368: Programming Massively Parallel Processors with CUDA
EECS 452: Advanced Computer Architecture
EECS 453: Parallel Architectures

Area 2. VLSI & CAD

EECS 353: Digital Microelectronics
EECS 355: ASIC and FPGA Design
EECS 357: Introduction to VLSI CAD
EECS 391: VLSI Systems Design
EECS 392: VLSI Systems Design Projects
EECS 393/493: Design and Analysis of High-Speed Integrated Circuits
EECS 459: VLSI Algorithmics

Area 3. Embedded Systems

EECS 332: Digital Image Analysis
EECS 346: Microprocessor System Design
EECS 347-1, 347-2: Microprocessor System Projects
EECS 390: Introduction to Robotics

BME 384: Biomedical Computing

Area 4. Algorithm Design and Software Systems

EECS 212: Mathematical Foundations of Computer Science
EECS 214: Data Structures and Data Management
EECS 321: Programming Languages
EECS 322: Compiler Construction
EECS 336: Design and Analysis of Algorithms
EECS 339: Introduction to Database Systems
EECS 343: Operating Systems
EECS 395: Introduction to the Theory of Computation

Fundamental EECS Courses

EECS 213: Introduction to Computer Systems
EECS 222: Fundamentals of Signals and Systems
EECS 223: Fundamentals of Solid State Engineering
EECS 224: Fundamentals of Electromagnetics and Photonics
EECS 225: Fundamentals of Electronics

Computer Engineering Capstone Design Requirement – one course

To satisfy the Department's Computer Engineering design capstone course requirements, students must elect to take at least one course from the following menu:

EECS 347-1: Microprocessor System Projects (346 is prerequisite)
EECS 362: Computer Architecture Projects (361 is prerequisite)
EECS 392: VLSI Design Projects (391 or 355 is prerequisite)

Upon completion of the design project, students are required to prepare a report using the template provided at
http://www.eecs.northwestern.edu/images/docs/forms/capstonedesign.pdf

COMPUTER ENGINEERING P/N POLICY STATEMENT:

Among the 16 departmental courses, the P/N option may only be used within the three technical electives that can be any 300-level course from science, mathematics, computer science, engineering or the areas (but beyond the required five courses from the area and the two fundamental EE courses). In addition, students may have no more than two P or D grades within the 16 departmental courses.

REQUIREMENT FOR GRADUATION In addition to McCormick graduation requirements, a student must receive a C– or better in EECS 202 and EECS 203 in order to continue in the Computer Engineering program.

3.4 Preferred Schedule for Computer Engineering

2013-14

	FALL	WINTER	SPRING
Freshman	Math 220 Chem 101 or Basic Science GenEng- 205-1 COMM ST 102 or 103	Math 224 Chem 102 or Basic Science GenEng - 205-2 IDEA 106-1	Math 230 EECS 203 (F,W) GenEng 205-3 IDEA 106-2
Sophomore	EECS 205 (W) Physics 135-2 GenEng - 205-4 SS/Hum	Physics 135-3 EECS 202 (F,S) EECS 211 (F,S) Math 234	EECS 214 EECS 221 (W) EECS 303 (F) SS/Hum
Junior Non Co-op	EECS Fund *(W,S) EECS 361(W) Basic Eng SS/Hum	Tech Elective Tech Elective EECS Fund* (F,S) SS/Hum	EECS 302 (F) Tech Elective Unres Elective SS/Hum
Senior Non Co-op	Tech Elective Tech Elective Unres Elective SS/Hum	Tech Elective CE Design (F,S) Unres Elective SS/Hum	Tech Elective Tech Elective Unres Elective Unres Elective
Junior Co-op	EECS Fund* (W,S) EECS 361(W) Basic Eng SS/Hum	EECS Fund (F,S) Tech Elective Tech Elective SS/Hum	Work
Pre-Senior Co-op	EECS 302 (S) Tech Elective Unres Elective SS/Hum	Work	Tech Elective Tech Elective Unres Elective SS/Hum
Senior Co-op	Work	Tech Elective CE Design (S) Unres Elective SS/Hum	Tech Elective Tech Elective Unres Elective Unres Elective

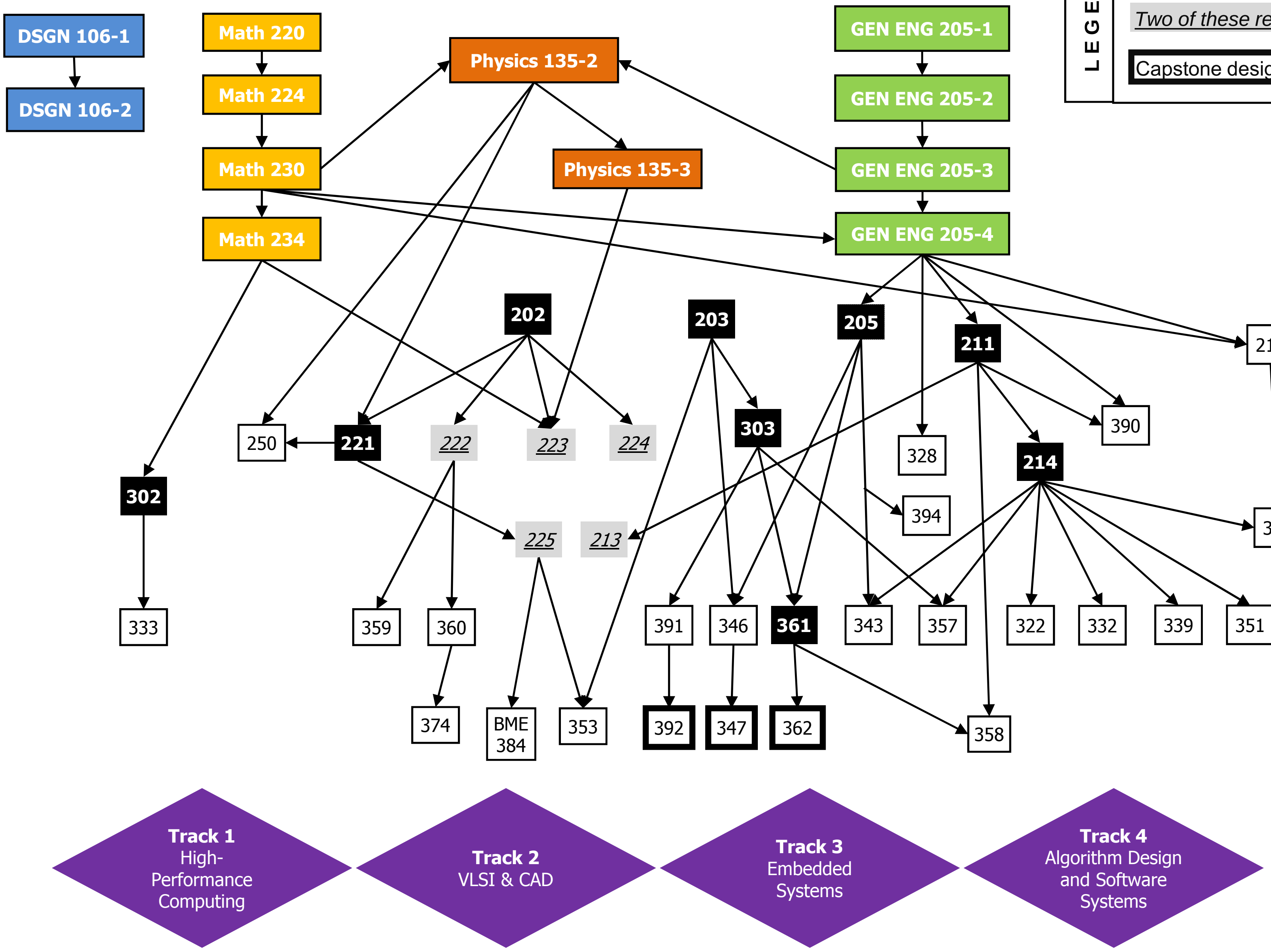
* Any two of the following EECS Fundamentals Courses: EECS 213, 222, 223, 224, 225.
Note that 221 and 223 must be taken before 225.

Updated April 4, 2013 RJ/ja

3.5 Computer Engineering Course Prerequisites

LEGEND

- Required courses
- *Two of these required*
- Capstone design



3.6 COMPUTER ENGINEERING STUDY PLAN

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
General Engineering, MathematiEEECS, Basic Science																		
G_E 205-1																		
G_E 205-2																		
G_E 205-3																		
G_E 205-4																		
IDEA 106-1																		
IDEA 106-2																		
Math 220																		
Math 224																		
Math 230																		
Math 234																		
Phy 135-2																		
Phy 135-3																		

Basic Engineering																		
EECS 202																		
EECS 203																		
EECS 211																		
EECS 302																		

Communications & Social Science/Humanities																		
G_C102 or 103																		

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Departmental Program																		
EECS 205																		
EECS 221																		
EECS 303																		
EECS 214																		
EECS 361																		

Capstone Design																		

Technical Electives: Tracks																		

Technical Electives: EE Fundamentals																		

Technical Electives: Remaining Three																		

Unrestricted																		

3.7 Conformance of the Computer Engineering Curriculum with ABET Guidelines

The following table indicates how the Computer Engineering curriculum conforms to the guidelines established by ABET Engineering Accreditation Commission.

Year & Quarter	Course (Department, Number, Title)	*Curricular Area (quarter course units)*			
		Math & Basic Sciences	Engineering Topics Check if Contains Design (√)	General Education	Other
1, Fall	Math 220, Calculus I	1			
1, Fall	Basic Science Elective	1			
1, Fall	GEN_ENG-205-1, Computational Methods and Linear Algebra	0.8	0.2		
1, Fall	General Communication 102 or 103			1	
1, Winter	Math 224, Calculus II	1			
1, Winter	Basic Science Elective 2	1			
1, Winter	GEN_ENG_205-2, Linear Algebra and Mechanics	0.5	0.5		
1,Winter	DSGN 106-1, Engineering Design and Communications I		0.5 √	0.5	
1, Spring	Math 230, Vectors and Partial Derivatives	1			
1, Spring	GEN_ENG-205-3, Dynamic System Modeling	0.8	0.2		
1, Spring	DSGN 106-2, Engineering Design and Communications II		0.5 √	0.5	
1, Spring	EECS 203, Intro to Computer Engineering		1 √		
2, Fall	EECS 205, Fundamentals of Computer System Software		1		
2, Fall	Physics 135-2, Electricity and Magnetism	1			
2, Fall	GEN_ENG-205-4, Differential Equations	0.9	0.1		
2, Fall	Hum/Social Science Elective 1			1	
2, Winter	EECS 202, Intro to Electrical Engineering		1		
2, Winter	Math 234 – Calculus and Analytical Geometry	1			
2,Winter	Physics 135-3, Waves	1			
2, Winter	EECS 211, Fundamentals of Programming		1 √		
2, Spring	EECS 221, Fund of circuits	0.25	0.75		
2, Spring	EECS 214 (formerly 311), Data Structures and Data Management	0.2	0.8		
2, Spring	EECS 303 Advanced Digital Logic Design		1 √		
2, Spring	Hum/Social Science Elective 2			1	
3, Fall	EECS Fundamental 1	0.25	0.75		

3, Fall	EECS 361 Computer Architecture I		1 √		
3, Fall	Basic Engineering elective		1		
3, Fall	Hum/Social Science Elective 3			1	
3, Winter	EECS Fundamental 2	0.25	0.75		
3, Winter	Technical Elective 1		1 √ *		
3, Winter	Unrestricted Elective 1			1	
3, Winter	Hum/Social Science Elective 4			1	
3, Spring	EECS 302, Probability	0.5	0.5		
3, Spring	Technical Elective 2		1√ *		
3, Spring	Unrestricted Elective 2			1	
3, Spring	Hum/Social Science Elective 5			1	
4, Fall	Technical Elective 3		1√ *		
4, Fall	Technical Elective 4		1√ *		
4, Fall	Social Science Elective 6			1	
4, Fall	Unrestricted Elective 3			1	
4, Winter	Technical Elective 5		1√ *		
4, Winter	Social Science Elective 7			1	
4, Winter	Capstone Design		1√		
4, Winter	Unrestricted Elective 4			1	
4, Spring	Technical Elective 6		1√ *		
4, Spring	Technical Elective 7		1√ *		
4, Spring	Technical Elective 8		1√ *		
4, Spring	Unrestricted Elective 5			1	
TOTALS-ABET BASIC-LEVEL REQUIREMENTS		12.45	21.55	14	
OVERALL TOTAL QUARTER UNITS FOR THE DEGREE : 48					
PERCENT OF TOTAL		26%	44.8%	29.2%	
Total must satisfy either credit hours or percentage	Minimum Quarter Units	12 Units	18 units		
	Minimum Percentage	25%	37.5 %		

*Many but not all technical electives contain engineering design.

4.0 Computer Science Undergraduate Curriculum

Bachelor of Science in Computer Science (McCormick CS)
Bachelor of Arts in Computer Science (Weinberg CS)
Majors in Computer Science (McCormick and Weinberg)
Minors in Computer Science (McCormick and Weinberg)
Second Major in Computer Science for ISP Students (Weinberg)

The purpose of this document is to explain the Computer Science (CS) Undergraduate Curriculum. CS is offered as a major within both McCormick and Weinberg, with identical in-major requirements. Additionally, minors are available in both schools, again with identical in-minor requirements.

Because of this dual-school nature, this document is structured differently from a typical McCormick (BS) or Weinberg (BA) curriculum. The document focuses on describing the elements of the major in a manner independent of the frameworks of both schools. This description is intended to explain the nature of the degree to students who may be unfamiliar with the frameworks, and to external readers. Appendix A then shows how the major curriculum is mapped into the McCormick framework to lead to a BS in CS, while Appendix B shows how it is mapped into the Weinberg framework to lead to a BA in CS. In both schools, second majors in CS are possible with identical requirements to those listed for the BS and BA. The second major in CS for ISP students has slightly different requirements, which are also explain in Appendix B.

The minor is a strict subset of the major, and is explained in detail in Appendix C, including how it maps in both McCormick and Weinberg.

Students should consult with their advisors in case of ambiguity or special cases.

1. Philosophy

Computer Science as a field grew out of Electrical Engineering, Mathematics, and Psychology over 50 years ago. It synthesized aspects of these fields and grew exponentially over the past half century, both in terms of the number of Computer Science practitioners, and its economic and social impact on the world. The field continues its exponential growth. In November, 2005, the U.S. Bureau of Labor Statistics estimated that almost 60% of new science and engineering jobs and net replacements in the United States through 2014 would be for computer specialists. The National Association of Colleges and Employers (NACE) 2010 survey found that Computer Science had the 5th highest average starting salary ($61,407) of any college degree.

Given this growth, and the sheer breadth and scale of the Computer Science enterprise in industry and academia, what are the goals of the Northwestern undergraduate degree? A Northwestern Computer Science graduate will

- Comprehend the breadth of Computer Science, its key intellectual divisions and questions, and its past and likely future impacts on engineering, science, medicine, business, and law;
- Approach problems from the algorithmic perspective, understanding the nature of and broad reach of computation and how to apply it abstractly;
- Approach problems from the systems perspective, understand the evolving layers of the software/hardware stack and how to use and extend them;
- Approach problems from the intelligence perspective, understanding how to make progress against seemingly intractable problems;
- Design and implement complex software systems, individually and as a team member; and
- Design and implement effective human-computer interfaces.

Additionally, Northwestern graduates also will have had the opportunity to broaden their education by taking advantage of Computer Science’s strong connections to Northwestern programs in Computer

Engineering, Cognitive Science, and the Learning Sciences. Northwestern graduates may have also participated in directed research.

A Northwestern graduate will be imminently employable in the computer and software industries, and well beyond, as skills such as these are widely sought after. Our program will also provide effective preparation for graduate studies in Computer Science.

2. Engineering or Liberal Arts

Northwestern offers Computer Science degrees within McCormick, the Engineering and Applied Sciences School, and Weinberg, the Arts and Sciences School. The Computer Science-specific elements/requirements of the two degrees are identical. The McCormick degree offers a background in engineering, while the Weinberg degree offers a background in liberal arts.

3. Prerequisite Graph and Schedule

A detailed prerequisite graph is available as a separate document, as is a typical schedule.

4. Components of the Curriculum

The Northwestern Computer Science Degree (the major) is composed of five distinct sets of requirements. Background requirements build up the student's engineering skills. The Core requirements represent essential knowledge for all computer scientists. The Breadth requirements provide exposure to every critical subfield of Computer Science. The Depth requirements provide the student with the opportunity to learn about two specializations in depth, leading to a project, and perhaps graduate courses and research. The Project requirement gives the student the experience of designing and building a complex software artifact. The minor in Computer Science consists of some of the Background requirements, all of the Core requirements, and some of the Breadth requirements.

4.1 Background

Background courses are those courses that fulfill the general requirements of the University and the School (i.e., McCormick or Weinberg), as well as non-CS courses that the faculty believe are foundational for or helpful for understanding Computer Science.
Because background courses are strongly dependent on the School, in this section we present the CS background requirements at a high level. Appendices A and B give detailed descriptions of required or recommended courses within the two Schools. We require students to take courses that teach the following:

- **Continuous Mathematics**. Students must learn univariate differential and integral calculus, multivariate differential calculus, and linear algebra.
- **Probability and Statistics**. Students must learn basic probability theory, and basic statistics, including descriptive statistics and hypothesis testing.
- **Physical and Life Sciences**. Students must meet the requirements of their school. We recommend that students choose physics and biology courses.
- **Social Sciences and the Humanities**. Students should acquire a firm grounding in these areas, ideally choosing courses that integrate into a theme.
- **Communication Skills**. Students must learn the fundamentals of effective written, graphical, and oral communication.

Students, who have never programmed before, in any language, are encouraged to take one of the following courses as part of their background:

- EECS 110 (C) – An Introduction to Programming for Non-majors using the C programming language. C is a widely used systems programming language.
- EECS 110 (Python) – An Introduction to Programming for Non-majors using the Python programming language. Python is a widely used scripting language.

Students are generally encouraged to take the Python option.

4.2 Core

The core courses reflect what the faculty expects every graduate to know. The core courses, all of which are required, consist of:

- EECS 101 – An Introduction to Computer Science for Everyone. This course should ideally be taken in the freshman year. This course can also be replaced with any course that would satisfy a breadth requirement.
- EECS 111 – Fundamentals of Computer Programming I. This course teaches principles and practices of computer programming using a functional programming language.
- EECS 211 – Fundamentals of Computer Programming II. This course teaches further principles and practices of computer programming using an imperative, object-oriented programming language.
- EECS 212 (*formerly 310*) – Discrete Mathematics. This course introduces the mathematics underlying much of Computer Science.
- EECS 213 – Introduction to Computer Systems. This course teaches how the computer system works, from the level of transistors to the level of distributed systems, with a particular focus on instruction set architecture, compilers, and operating systems.
- EECS 214 (*formerly 311*) – Introduction to Data Structures. This course introduces software structures and algorithms for storing, accessing, and transforming information and their implementation

4.3 Breadth (Essential Areas)

The breadth courses reflect the areas of Computer Science that faculty believe you should be exposed to. A student must take one course in each of the following areas. Each course can only count for a single area. It is possible to use EECS 399 (independent study) courses within breadth, with advisor and committee approval. This is rare and requires a petition.

Theory: Theory courses study the nature of computation, the nature of computational problems, and the design of algorithms for solving these problems efficiently. The following courses are appropriate for satisfying the theory breadth requirement. The courses with asterisks are especially recommended.

- * EECS 335 – Introduction to the Theory of Computation
- * EECS 336 – Design and Analysis of Algorithms

Systems: Systems courses study the layers of the hardware/software stack, and the design, implementation, and evaluation of complex software systems, including computer security. The following courses are appropriate for satisfying the systems breadth requirement. The courses with asterisks are especially recommended.

- * EECS 321 – Programming Languages
- * EECS 322 – Compiler Construction
- * EECS 339 – Introduction to Databases
- * EECS 340 – Introduction to Networking
- * EECS 343 – Operating Systems
- * EECS 354 – Network Penetration and Security
- EECS 303 – Digital Logic Design
- EECS 345 – Distributed Systems
- EECS 346 – Microprocessor Systems Design
- EECS 350 – Introduction to Security
- EECS 369 - Introduction to Sensor Network
- EECS 395/495 - Data Science
- EECS 395/495 - (Special Topics) Embedded Systems
- EECS 395/495 - Kernel and Other Low-level Software Development
- EECS 399 - Independent Study

- EECS 440 – Advanced Networking
- EECS 441 – Resource Virtualization
- EECS 450 – Internet Security

Artificial Intelligence: Those in the field of Artificial Intelligence seek scientific understanding of the mechanisms underlying thought and intelligent behavior and look to embody these mechanisms in machines. Intelligent learning environments, computer games, and music retrieval systems are some applications of AI technology. The following courses are appropriate for satisfying the AI breadth requirement. The courses with asterisks (*) are especially recommended.

- * EECS 325 – AI Programming
- * EECS 337 – Semantic Information Processing
- * EECS 344 – Design of Computer Problem Solvers
- * EECS 348 – Introduction to AI
- * EECS 349 – Machine Learning
- EECS 360 – Introduction to Feedback Systems
- EECS 371 – Knowledge Representation
- EECS 372/472 - Designing and Constructing Models with Multi-Agent Languages
- EECS 395/495 – Knowledge, Representation & Reasoning for Game Characters
- EECS 395/495 – Machine Learning: Foundations, Applications, and Algorithms

Interfaces: Courses in this area study the human-computer interface, including computer graphics and multimedia processing. The following courses are appropriate for satisfying the interface breadth requirement. The courses with asterisks are especially recommended.

- * EECS 330 – Human-Computer Interaction
- * EECS 351 – Introduction to Computer Graphics
- * EECS 352 – Machine Perception of Music
- * EECS 370 – Computer Game Design
- EECS 332 – Introduction to Computer Vision
- EECS 395 - Game Design and Development
- EECS 395/495 - Biometrics
- EECS 395/495 - Computational Photography Seminar
- EECS 395/495 – Design, Technology, and Research
- EECS 395/495 – Game Development Studio
- EECS 395/495 – (Special Topics) Innovation in Journalism & Technology
- EECS 395/495 – Geospatial Vision and Visualization
- EECS 395/495 – Intermediate Computer Graphics
- EECS 395/495 – Introduction to Computational Photography
- EECS 395/495 (LRN_SCI 451) – Tangible Interaction Design and Learning (Horn)

Software Development: Courses in this area provide opportunities to experience larger-scale software development in teams. Scaling software development is an important challenge. The following courses are appropriate for satisfying the software development breadth requirement. The courses with asterisks are especially recommended.

- * EECS 338 – Practicum in Intelligent Information Systems
- * EECS 394 – Software Project Management
- * EECS 395- Software Construction
- EECS 395/495 - Game Development Studio
- * EECS 473- 1,2 NUvention: Web (must complete both quarters; 473-1 may then count toward the software development breadth requirement and 473-2 as a project course)
-

4.4 Depth

Students are expected to acquire depth in two of the areas listed below. Students should consult with their advisor to choose the two most appropriate areas of study for them. A total of six courses should be taken in the two chosen areas, with three courses in each area.

A student may petition to be allowed to focus on a single area, if desired. This petition needs to be approved by the student's advisor, but we intend for this to be straightforward.

A student may petition to use independent study courses (EECS 399s) in depth areas that permit them. This petition needs to be approved by the student's advisor and committee. Again, we intend this to be simple.

A student interested in defining his/her own depth area is encouraged to provide a justification for the proposed set of depth courses to his/her advisor, who may in turn petition the Computer Science Curriculum Committee for its approval.

The Computer Science Curriculum Committee will meet once per year, in the fall quarter, to consider additions, deletions, or changes to the depth areas and the list of appropriate courses for each area.

Theory: Theory courses study the nature of computation, the nature of computational problems, and the design of algorithms for solving these problems efficiently. The following courses are considered appropriate for satisfying the theory depth area. The courses with asterisks are especially recommended.

- * EECS 335 – Introduction to the Theory of Computation
- * EECS 336 – Design and Analysis of Algorithms
- * EECS 457 – Advanced Algorithms
- EECS 395/495 – Algorithmic Techniques for Bioinformatics
- EECS 395/495 – Computational Geometry
- EECS 395/495 – Introduction to Computational Learning Theory
- EECS 399 – Independent Study
- EECS 428 – Information Theory
- EECS 459 – VLSI Algorithmics

Systems: In the depth area of systems, a student will learn about the issues and principles involved in the design, implementation, measurement, and analysis of the complex software and software/hardware systems upon which applications are built. Students will learn how computer systems work, from the level of the hardware to the level of worldwide distributed communication and computation. The principles and issues involved have broad utility and the skills that student will learn are in high demand.

- * EECS 321 – Programming Languages
- * EECS 322 – Compiler Construction
- * EECS 339 – Introduction to Databases
- * EECS 340 – Introduction to Computer Networking
- * EECS 343 – Operating SystemsI
- EECS 345 – Distributed Systems
- EECS 350 – Introduction to Computer Security
- EECS 354 – Network Penetration and Security
- EECS 368/468 – Programming Massively Parallel Processors with CUDA
- EECS 369 – Introduction to Sensor Networks
- EECS 395/495 - Advanced Networking
- EECS 395/495 – The Art of Multicore Concurrent Programming
- EECS 395/495 – Code Analysis and Transformation
- EECS 395/495 - Concurrent Programming in Rust
- EECS 395/495 – Data Science
- EECS 395/495 - Digital Forensics and Incident Response
- EECS 395/495 – (Special Topics) Embedded Systems
- EECS 395, 495: Internet of Things
- EECS 395/495 - Kernel and Other Low-level Software Development

- EECS 395/495 – Networking Problems in Cloud Computing
- EECS 395/495 – Wireless and Mobile Health (mHealth)
- EECS 399 – Independent Study
- EECS 440 – Advanced Networking
- EECS 441 – Resource Virtualization
- EECS 444 – Networking Problems in Cloud Computing
- EECS 450 – Internet Security

Artificial Intelligence: Those in the field of Artificial Intelligence seek scientific understanding of the mechanisms underlying thought and intelligent behavior and look to embody these mechanisms in machines. Intelligent learning environments, computer games, and music retrieval systems are some applications of AI technology. The following courses are appropriate for depth in AI. The courses with asterisks are especially recommended.

- * EECS 301– Introductory Robotics Laboratory
- * EECS 325 – AI Programming
- * EECS 337 – Semantic Information Processing
- * EECS 344 – Design of Computer Problem Solvers
- * EECS 348 – Introduction to AI
- * EECS 349 – Machine Learning
- * EECS 371 – Knowledge Representation
- EECS 372/472 - Designing and Constructing Models with Multi-Agent Languages
- EECS 395/495 – Computational Geometry
- EECS 395/495 – (Special Topics) Innovation in Journalism & Technology
- EECS 395/495 – Introduction to Computational Learning Theory
- EECS 395/495 – Knowledge, Representation & Reasoning for Game Characters
- EECS 395/495 – Special Topics in Machine Learning
- EECS 395/495 – Special Topics in Swarms and Multi-robot Systems
- EECS 399 – Independent Study
- EECS 433 – Statistical Pattern Recognition
- EECS 469 – Machine Learning and Artificial Intelligence for Robotics
- EECS 474 – Probabilistic Graphical Models

Interfaces: Courses in this area study the human-computer interface, including computer graphics and multimedia processing. The following courses are appropriate for satisfying depth in interfaces. The courses with asterisks are especially recommended.

- * EECS 330 – Human-Computer Interaction
- * EECS 351 – Introduction to Computer Graphics
- * EECS 352 – Machine Perception of Music
- * EECS 370 – Computer Game Design
- EECS 332 – Digital Image Analysis
- EECS 395 – Game Design and Development
- EECS 395/495 – Biometrics
- EECS 395/495 – Cognitive Simulation for Virtual Characters
- EECS 395/495 - Computational Photography Seminar
- EECS 395/495 – Design, Technology, and Research
- EECS 395/495 – Game Development Studio
- EECS 395/495 – Geospatial Vision and Visualization
- EECS 395/495 – (Special Topics) Innovation in Journalism & Technology
- EECS 395 – Intermediate Computer Graphics
- EECS 395/495 – Introduction to Computational Photography
- EECS 395/495 (LRN_SCI 451) – Tangible Interaction Design and Learning (Horn)
- EECS 399 – Independent Study

Security: This area focuses on the fundamental principles of computer and communication security as well as their implications and applications in various domains of information technology. The following courses satisfy the security depth area. The courses with asterisks are especially recommended.

- EECS 350 – Introduction to Computer Security
- EECS 354 – Network Penetration and Security
- EECS 395/495 – Digital Forensics and Incident Response
- EECS 444 – Networking Problems in Cloud Computing

4.5 Project Requirement

In both the old and new Computer Science Curricula, students must complete at least two quarters of project-oriented classes. The best option is a two-quarter independent study project (EECS 399) with a faculty member. However, it is also possible to satisfy the requirement by taking two courses with extensive project work. Students must seek approval of (e.g. petition) their advisor for the option chosen.

Possible ways of satisfying the Project requirement include:

- A single two quarter EECS 399 project
- Two independent single quarter EECS 399 projects
- One EECS 399 project and one project course
- Two project courses

A list of courses that satisfy the Project Course requirement can be found by searching for the subset "CS Project Courses" on the course descriptions page of the EECS website: http://www.mccormick.northwestern.edu/eecs/courses/computer-science-curriculum/cs-project-course.html

Designating a course as a project course is ultimately at the discretion of the course director. As guidance, the important aspects of a project course are that it be:

- open ended
- (relatively) self-directed
- long, as a fraction of the time spent on it in a quarter

That is, a course that has a heavy workload is not necessarily a project course. For example, a course where the work is divided up into weekly or biweekly, stand-alone, circumscribed chunks is not a project course, no matter the magnitude of the combined chunks.

At least one (or possibly both) of the project courses must be implementation-based project courses.

4.6 Restrictions

Courses may not be double counted within the major program. Students may take EECS 399 and EECS 338 no more than a total of four times. A petition is required for taking these courses additional times.

Appendix A. Mapping to the McCormick Framework

A.1 Mathematics: Students must take the following four courses.

- MATH 220 – Calculus I
- MATH 224 – Calculus II
- MATH 230 – Calculus III
- EECS 212 *(formerly EECS 310)* – Discrete Mathematics

A.2 Engineering Analysis: Students must take the following four courses.

- GEN_ENG 205-1 – EA-1 (Linear Algebra)
- GEN_ENG 205-2 – EA-2 (Mechanics)
- GEN_ENG 205-3 – EA-3 (Dynamics and Differential Equations)
- EECS 111 – Fundamentals of Computer Programming I

Note that students do not need to take EA-4 for the CS degree. However, students may elect to do so if they are also interested in other degrees in McCormick.

A.3 Engineering Design and Communication: Students must take the following three courses.

- IDEA 106-1 and ENGLISH 106-1 – EDC-1
- IDEA 106-2 and ENGLISH 106-2 – EDC-2
- COMM ST 102 Public Speaking or PERF ST 103 Analysis and Performance of Literature

B.4 Basic Sciences: Students must take four courses that satisfy McCormick requirements. We recommend that students focus on Physics and Biology courses for maximum utility in their Computer Science Degree. We recommend that students take all of the following courses:

- PHYSICS 135-2 General Physics: Electromagnetics
- PHYSICS 135-3 General Physics: Wave Phenomena
- PHYSICS 335 Modern Physics

and choose one of these courses:

- BIOL SCI 215 Genetics and Molecular Biology
- CHEM_ENG 275 Molecular and Cell Biology for Engineers

Note that PHYSICS 135-1 – Mechanics is not suitable for satisfying the basic sciences requirement in engineering. However, the material in it is covered in EA-2. A common sequence for satisfying the basic sciences requirement in Computer Science is EA-2, PHYSICS 135-2, PHYSICS 135-3, PHYSICS 335, and BIOL SCI 215. This sequence will give you a firm grounding in classical and modern physics, upon which present and possibly future computers are based, and in genetics and evolution, where computational approaches to biology and biological approaches to computation abound.

A.5 Basic Engineering: Students must take five courses that meet the following requirements.

- EECS 211 – Fundamentals of Computer Programming II must be taken. This course is in the area “Computer Programming”
- EECS 302 – Probabilistic Systems and Random Signals, IEMS 201 – Introduction to Statistics, or IEMS 303 – Statistics I must be taken. These courses are in the area “Probability, Statistics, and Process Control”.
- Three additional courses from McCormick’s Basic Engineering List (3 courses from at least two areas, excluding those covered by EECS 211 and 302). We recommend choosing from the following courses to meet this requirement:
    - EECS 202 – Introduction to Electrical Engineering (Area “Electrical Science”)

  - EECS 203 – Introduction to Computer Engineering (Area “Computer Architecture and Numerical Methods”)
  - EECS 222 – Fundamentals of Signals and Systems (Area “Electrical Science”)
  - EECS 328 – Numerical Methods for Engineers (Area “Computer Architecture and Numerical Methods”)

A.6. Social Sciences and Humanities: Students must take seven courses that meet the theme requirements of McCormick.

A.7 Unrestricted Electives: Students must take five courses.

Students who have never programmed before, in any language, are encouraged to use one of the unrestricted electives to take one of the following courses before taking EECS 111:

- EECS 110 (C) – An Introduction to Programming for Non-majors using the C programming language. C is a widely used systems programming language.
- EECS 110 (Python) – An Introduction to Programming for Non-majors using the Python programming language. Python is a widely used scripting language.

Students are generally encouraged to take the Python option. Note that EECS 110 is not a course within the CS major program.

A.8 Major Program (16) : Students must take 16 courses that meet the following requirements. No courses may be taken pass/fail. No courses may be double-counted within the major program.

A.8.1 Core Courses (3): Students must take the following three core courses as part of the major program. The core courses, including those mapped outside the major program, are described in detail in Section 4.2.

- EECS 101 – EECS 101 – An Introduction to Computer Science for Everyone. This course should ideally be taken in the freshman year. This course can also be replaced with any course that would satisfy a breadth requirement
- EECS 213 – Introduction to Computer Systems
- EECS 214 *(formerly EECS 311)* – Introduction to Data Structures

A.8.2 Breadth Courses (5): Students must take five courses to satisfy the Breadth requirement, one in each area described in Section 4.3.

A.8.3 Depth Courses (6): Students must take six courses to satisfy the Depth requirement as described in Section 4.4.

A.8.4 Project Courses (2): Students must take two courses that satisfy the Project component of the curriculum as described in Section 4.5.

Appendix B. Mapping to the Weinberg Framework

B.1 Requirements to be met outside of the major program.

- Mathematics: MATH 220, 224, 230, and 240 are required
- Probability and Statistics: STATS 210 or MATH 310-1 (Probability and Stochastic Processes) or a 5 on the AP Statistics exam are required
- Physical and Life Sciences: Students must satisfy the Natural Sciences distribution requirement. We recommend choosing from courses such as PHYSICS 135-1, 135-2, 135-3, 335; and BIOL SCI 210-1
- Social Sciences and Humanities: Covered by Weinberg's distributional requirements
- Communication Skills: Covered by Weinberg's distributional requirements

Students who have never programmed before, in any language, are encouraged to take one of the following courses before taking EECS 111:

- EECS 110 (C) – An Introduction to Programming for Non-majors using the C programming language. C is a widely used systems programming language.
- EECS 110 (Python) – An Introduction to Programming for Non-majors using the Python programming language. Python is a widely used scripting language.

Students are generally encouraged to take the Python option.

B.2 Requirements to be met within the major program (19). No courses may be taken pass/fail. No courses may be double-counted within the major program.

B.2.1 Core Courses (6): Students must take the following six core courses as part of the major program. The core courses are described in detail in Section 4.2.

- EECS 101 – An Introduction to Computer Science for Everyone. This course should ideally be taken in the freshman year. This course can also be replaced with any course that would satisfy a breadth requirement
- EECS 111 – Fundamentals of Computer Programming I
- EECS 211 – Fundamentals of Computer Programming II
- EECS 212 *(formerly EECS 310)* – Discrete Mathematics
- EECS 213 – Introduction to Computer Systems
- EECS 214 *(formerly EECS 311)* – Introduction to Data Structures

B.2.2 Breadth Courses (5): Students must take five courses to satisfy the Breadth requirement, one in each area described in Section 4.3.

B.2.3 Depth Courses (6): Students must take six courses to satisfy the Depth requirement as described in Section 4.4.

B.2.4 Project Courses (2): Students must take two courses that satisfy the Project component of the curriculum as described in Section 4.5.

*****B.3. Second major in CS for ISP students:** The requirements for Background (B.1) are met via the participation in the ISP program. The Core requirements (B.2.1) and Breadth requirements (B.2.2) are exactly as described above. There is no Depth requirement (B.2.3 is eliminated). The Project requirement (B.2.4) must be met with 2 quarters of ISP 398 or 2 quarters of EECS 399. In either case, the Project requirement must be approved by advisors both in ISP and in CS.

Appendix C. Minor in Computer Science

The minor in Computer Science is available in both McCormick and Weinberg. Requirements within the minor are identical across the schools.

C.1 Overview of Requirements

The requirements for the minor are a strict subset of those of the major. This makes it easy for students to convert from a minor to a major, and (in the early stages) from a major to a minor with no "lost" courses. The requirements are as follows:

- **Background:** Students must satisfy the math requirements of the major.
- **Core:** Students complete the same core requirements as in the major
- **Breadth:** Students complete one course in each of three Breadth areas (compared to five in the major)
- No courses may be taken pass/fail.

While the intent to pursue a minor may be declared at any point (the earlier the better), the petition to receive the minor must be completed and submitted for approval (McCormick Office for Undergrads) in the beginning of the final quarter of studies.

C.2 Mapping to McCormick (9 courses)

C.2.1 Background: The standard McCormick Math requirements (MATH 220, 224, 230) and standard EA requirements (specifically GEN_ENG 205-1 (EA-1) are sufficient. No additional courses need be taken.

C.2.2 Core Courses (6): Students must take the following six core courses as part of the major program. The core courses are described in detail in Section 4.2.

- EECS 101 – An Introduction to Computer Science for Everyone. This course should ideally be taken in the freshman year. This course can also be replaced with any course that would satisfy a breadth requirement
- EECS 111 – Fundamentals of Computer Programming I
- EECS 211 – Fundamentals of Computer Programming II
- EECS 212 *(formerly EECS 310)* – Discrete Mathematics
- EECS 213 – Introduction to Computer Systems
- EECS 214 *(formerly EECS 311)* – Introduction to Data Structures

C.2.3 Breadth Courses (3): Students must take one course in each of three different Breadth areas (specified in Section 4.3).

C.3 Mapping to Weinberg (9-13 courses)

C.3.1. Background: Students should complete Math 220, 224, 230, and 240, or their equivalent. Note that some of these courses may count toward other requirements.

C.3.2 Core Courses (6): Students must take the following six core courses as part of the major program. The core courses are described in detail in Section 4.2.

- EECS 101 – An Introduction to Computer Science for Everyone. This course should ideally be taken in the freshman year. This course can also be replaced with any course that would satisfy a breadth requirement
  - This course may also count toward the Social and Behavioral Sciences Distribution Area
- EECS 111 – Fundamentals of Computer Programming I
  - This course may also count toward the Formal Studies Distribution Area
- EECS 211 – Fundamentals of Computer Programming II
- EECS 212 *(formerly EECS 310)* – Discrete Mathematics
- EECS 213 – Introduction to Computer Systems
- EECS 214 *(formerly EECS 311)* – Introduction to Data Structures

C.3.3 Breadth Courses (3): Students must take one course in each of three Breadth areas. Breadth areas and their approved courses are listed in Section 4.3.

Requirements in a Nutshell

Applies to both McCormick (BS) and Weinberg (BA)
See www.eecs.northwestern.edu for detailed curriculum document

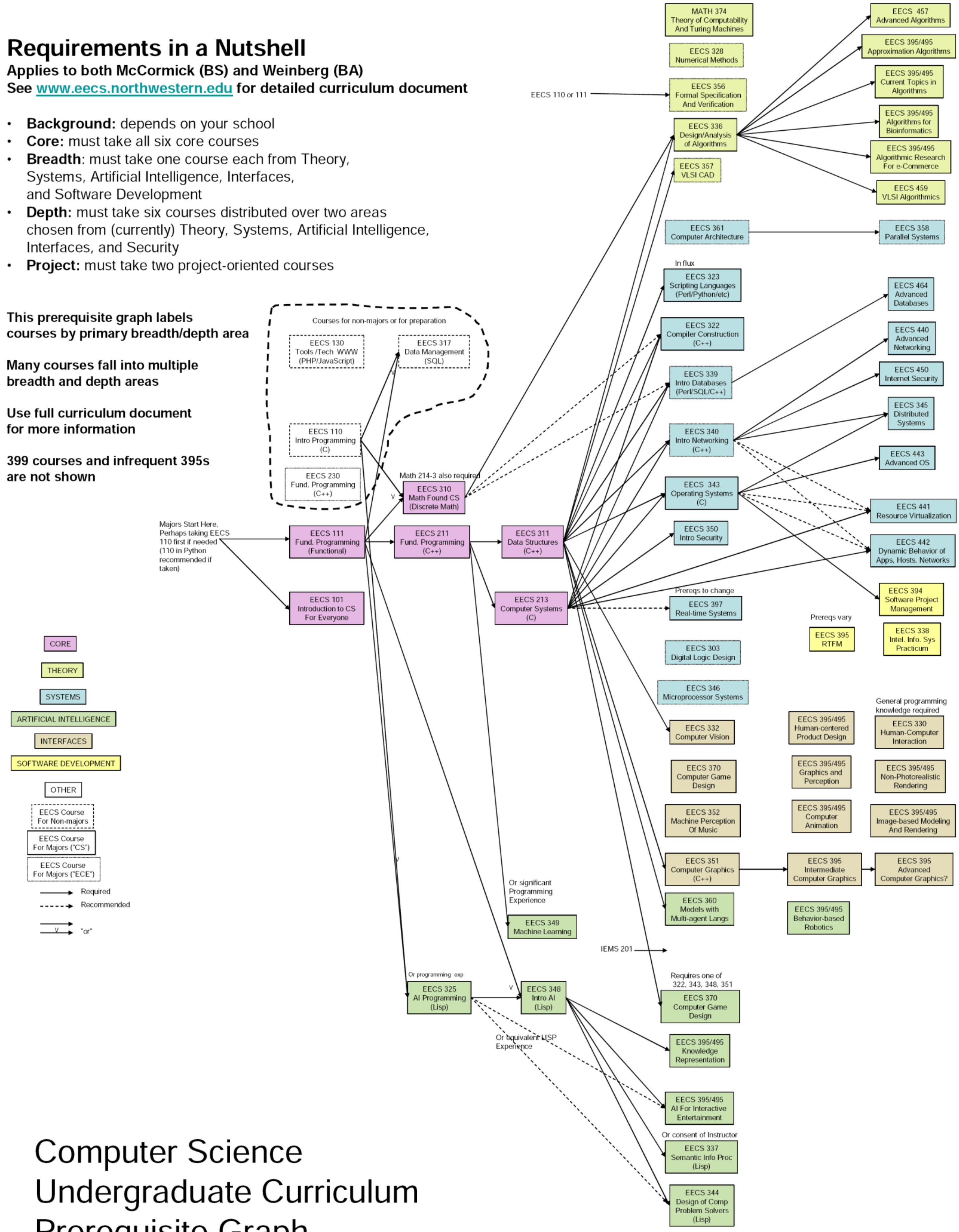
- **Background:** depends on your school
- **Core:** must take all six core courses
- **Breadth**: must take one course each from Theory, Systems, Artificial Intelligence, Interfaces, and Software Development
- **Depth:** must take six courses distributed over two areas chosen from (currently) Theory, Systems, Artificial Intelligence, Interfaces, and Security
- **Project:** must take two project-oriented courses

This prerequisite graph labels courses by primary breadth/depth area

Many courses fall into multiple breadth and depth areas

Use full curriculum document for more information

399 courses and infrequent 395s are not shown



Courses for non-majors or for preparation

EECS 130 Tools /Tech WWW (PHP/JavaScript)

EECS 317 Data Management (SQL)

EECS 110 Intro Programming (C)

EECS 230 Fund. Programming (C++)

Majors Start Here, Perhaps taking EECS 110 first if needed (110 in Python recommended if taken)

EECS 111 Fund. Programming (Functional)

EECS 101 Introduction to CS For Everyone

Math 214-3 also required

EECS 310 Math Found CS (Discrete Math)

EECS 211 Fund. Programming (C++)

EECS 311 Data Structures (C++)

EECS 213 Computer Systems (C)

MATH 374 Theory of Computability And Turing Machines

EECS 328 Numerical Methods

EECS 110 or 111

EECS 356 Formal Specification And Verification

EECS 336 Design/Analysis of Algorithms

EECS 357 VLSI CAD

EECS 457 Advanced Algorithms

EECS 395/495 Approximation Algorithms

EECS 395/495 Current Topics in Algorithms

EECS 395/495 Algorithms for Bioinformatics

EECS 395/495 Algorithmic Research For e-Commerce

EECS 459 VLSI Algorithmics

EECS 361 Computer Architecture

EECS 358 Parallel Systems

In flux

EECS 323 Scripting Languages (Perl/Python/etc)

EECS 322 Compiler Construction (C++)

EECS 339 Intro Databases (Perl/SQL/C++)

EECS 340 Intro Networking (C++)

EECS 343 Operating Systems (C)

EECS 350 Intro Security

Prereqs to change

EECS 397 Real-time Systems

EECS 303 Digital Logic Design

EECS 346 Microprocessor Systems

EECS 464 Advanced Databases

EECS 440 Advanced Networking

EECS 450 Internet Security

EECS 345 Distributed Systems

EECS 443 Advanced OS

EECS 441 Resource Virtualization

EECS 442 Dynamic Behavior of Apps, Hosts, Networks

EECS 394 Software Project Management

Prereqs vary

EECS 395 RTFM

EECS 338 Intel. Info. Sys Practicum

EECS 332 Computer Vision

EECS 395/495 Human-centered Product Design

General programming knowledge required

EECS 330 Human-Computer Interaction

EECS 370 Computer Game Design

EECS 395/495 Graphics and Perception

EECS 395/495 Non-Photorealistic Rendering

EECS 352 Machine Perception Of Music

EECS 395/495 Computer Animation

EECS 395/495 Image-based Modeling And Rendering

EECS 351 Computer Graphics (C++)

EECS 395 Intermediate Computer Graphics

EECS 395 Advanced Computer Graphics?

EECS 360 Models with Multi-agent Langs

EECS 395/495 Behavior-based Robotics

Or significant Programming Experience

EECS 349 Machine Learning

IEMS 201

Or programming exp

EECS 325 AI Programming (Lisp)

EECS 348 Intro AI (Lisp)

Or equivalent LISP Experience

Requires one of 322, 343, 348, 351

EECS 370 Computer Game Design

EECS 395/495 Knowledge Representation

EECS 395/495 AI For Interactive Entertainment

Or consent of Instructor

EECS 337 Semantic Info Proc (Lisp)

EECS 344 Design of Comp Problem Solvers (Lisp)

CORE

THEORY

SYSTEMS

ARTIFICIAL INTELLIGENCE

INTERFACES

SOFTWARE DEVELOPMENT

OTHER

EECS Course For Non-majors

EECS Course For Majors ("CS")

EECS Course For Majors ("ECE")

Required

Recommended

"or"

Computer Science Undergraduate Curriculum Prerequisite Graph

5. Electives and Non-technical Courses

In addition to the required math, science, and engineering courses described in each of the majors, all students take a number of non-technical and elective courses to satisfy the McCormick School requirements.

Social Science - Humanities Requirements

Students must take seven courses chosen according to either of the following two options:

Option A. At least two courses must be chosen in each of three areas:

Social and Behavioral Science (SBS)
Historical Studies and Values (HSV)
Fine Arts, Language and Literature (FALL)

Of the seven courses, only three A-level introductory courses may be presented and three courses must be thematically related to provide depth.

Option B. Five of the seven courses must clearly be thematically related. For breadth, no more than five courses may come from a single area. The courses taken for a student's Social Science/Humanities requirement must be approved in advance by the McCormick Humanities Panel. Theme declaration forms for either option are available in either the Undergraduate Records Office (L269) or the Freshman Program Office (L275). Once filled out and signed by a student's advisor, they should be returned to one of these offices for approval.

Unrestricted Electives

Unrestricted electives can be selected from any course offered for credit in the University. These are very valuable in permitting the student to concentrate in a particular area. For example, the student may utilize some or all of these in conjunction with the seven Social Science / Humanities Requirements to achieve an in-depth undergraduate preparation in the Social Science / Humanities Requirements area. As a second example, the student may utilize the unrestricted electives in conjunction with the technical electives to attain an undergraduate specialization in a particular area of Electrical Engineering and Computer Science. With suitable advanced placement credit, summer work, or a fifth year, it is possible in this manner to structure a curriculum that achieves two B.S. Degrees, one in Electrical Engineering and one in Computer Engineering. Students should interact closely with their advisors in planning their unrestricted electives to achieve the maximum benefit from their undergraduate program.

Communications Requirement

All McCormick School students are required to develop proficiency in writing and speaking before graduation, since effective communication is essential in any career in engineering, management, or academia. Written communication is stressed in the Engineering Design and Communication sequence. Students must also take a course in oral communication from this list:

General Communication 102 - Public Speaking
General Communication 103 - Analysis and Performance of Literature

6. Special Programs, Honors Programs

Concurrent BS/MS and BA/MS Programs

eecs.northwestern.edu/overview-of-our-undergraduate-programs/concurrent-bs-ms-ba-ms
Northwestern undergraduates have the opportunity to pursue an MS degree concurrently with their Bachelor's degree.

The EECS department offers BS/MS and BA/MS programs that lead to a master's degree in Computer Science (CS), or a master's degree in Electrical and Computer Engineering (ECE). These programs, offered through McCormick, but available to Northwestern undergraduates in both the Weinberg School and the McCormick School, allow talented undergraduates to undertake graduate level courses and to engage in research while they are completing their undergraduate requirements. Students completing an MS along with a BS or BA degree are likely to have greater career opportunities.

Co-Operative Engineering Education Program

mccormick.northwestern.edu/career-development/programs/co-op/

Co-operative engineering education is designed to provide alternate periods of industrial experience and classroom work for undergraduate students in all departments of engineering and applied science. During 18 months of industrial employment, a student is afforded the opportunity of applying theory while gaining practical experience. The perspective gained enables students to develop an understanding of the responsibilities of their future professional career.

Students in good academic standing normally elect the Co-op program in the Fall of the Sophomore year. The coordinator and career counselor make every effort to secure interviews for the student, with the long range goal of obtaining a cooperative work assignment related to the student's professional objectives.

Generally, the first work experience for Co-op students occurs during the summer between their Sophomore and Junior years. Co-op experience for Junior transfer students and others with two years of academic credit begins in the spring of their Junior year. Students are required to register for their work quarters, but no tuition or fee is charged. While no academic credit is given for Co-op, special BS/MS programs may utilize co-op experience as the basis for undergraduate projects and master's theses.

Although emphasis is placed upon the experience gained from Co-op work rather than upon the income, Co-op students may earn a sizable portion of their educational expenses.

If necessary, special schedules can be worked out with the help of the student's academic advisor that will enable the student to observe individual academic requirements as well as Co-op. These include four-year Co-op programs for students with advanced placement, and combined BS/MS programs. A student may be enrolled simultaneously in the Co-op program and in the Naval Reserve Officers Training Corps.

In addition to the academic degree, the faculty of the McCormick School awards the Co-op student a certificate in recognition of successful completion of the Co-operative Engineering Education Program.

McCormick School of Engineering & Applied Science Scholars Program

A high school student admitted into this program is almost immediately involved in the research program of an active faculty. Generally, the student will engage in research both during the academic year and during the summer, when he or she would be supported by the faculty or the McCormick School.

Students in this program follow an accelerated academic curriculum, capitalizing upon advanced placement credits. Depending upon the advanced standing and record at Northwestern the student might be admitted to the Graduate School in the Ph.D. program as early as the third academic year. Support for the student (full tuition and stipend) would then be provided by the University through a Cabell or Murphy Fellowship.

This program provides an opportunity for outstanding high school students to obtain their Ph.D. degrees in Electrical Engineering and Computer Science in as little as six years after their high school graduation. The students are provided with continuous opportunities to interact directly with active researchers and to publish in recognized scientific journals at an early age.

Undergraduate Honors Program

mccormick.northwestern.edu/undergraduates/programs/honors_and_combined_programs/undergraduate_honors_programs

A 4-year or co-op student with a good scholastic record may be admitted to the Undergraduate Honors Program anytime during the junior or pre-senior year. At the time of admission, the student must have a cumulative grade point average of 3.5 or better. Any student who becomes eligible will be so notified by the Dean.

An honors student participating in the program must complete at least three units of approved advanced study (including courses normally accepted at the graduate level, which most 300 level EECS technical electives are) with an average grade of B or better, and complete an extended independent study project (at least two quarters on the same topic) leading to an acceptable report.

Successful completion of the Honors program will be noted on the student's transcript. Recognition will also be given in the commencement program. If a student's individually evaluated performance is not judged to meet the standards of success, the student will receive course grades and credit as earned.

Business Enterprise Certificate

mccormick.northwestern.edu/undergraduates/curriculum/non_degree_programs/business_enterprise_certificate

Purpose: Although a significant percentage of our students ultimately pursue an advanced business degree, they will still spend several years in an organization before going back to school. This Certificate program is aimed at those who are intent upon having a career in business but who want to improve their ability to make a contribution soon as possible after finishing their McCormick degree. This educational experience gives them the orientation to make that possible. Read more on the McCormick website link, provided above.

Certificate in Engineering Design

www.idea.northwestern.edu/certificate/

Purpose: The certificate in engineering design program helps McCormick undergraduates develop a set of design skills that will prove valuable in their careers. The program focuses on innovative engineering design in a team-based, cross-disciplinary setting. "Innovative design" here implies both identifying and solving real-world problems. Read more on the McCormick website link, provided above.

7. Student Advising

Each student is assigned a faculty member as an advisor. Students meet with their advisors on a regular basis at least once each quarter. Advisors help students select appropriate courses to satisfy both the Departmental and McCormick requirements, and the student's own interests. In the Spring Quarter, the student and advisor design a tentative schedule of courses for the following academic year. In each of the remaining quarters, the advisor and student discuss the student's progress and verify or modify the schedule for the next quarter. Students are also invited to meet with their advisors at any time during the year to discuss academic and career goals.

In addition to individual student/advisor meetings, the Department as a whole has meetings with faculty and students. All of the faculty and undergraduate students are invited to attend and discuss academic and other questions, and to become acquainted in a less formal atmosphere. Advisors are generally assigned by the department; however, students can request a change of advisor by filling out a form in the Undergraduate Records Office.

The Dean may require a student having academic difficulties to meet with his or her advisor to discuss those difficulties.

Students who are interested in Premedical Studies should contact the Health Professions Advisors at the University Academic Advising Center (UAAC). The web address is: http://www.northwestern.edu/advising-center, or call 847-467-4281 for an appointment. Open Q&A sessions take place every Tuesday and Thursday afternoon.

8. Department Faculty

Academic Faculty

Brenna Argall, Assistant Professor; Ph.D., Carnegie Mellon University
Joint with Dept of Physical Medicine and Rehab (RIC)
Research Interests: robot autonomy and low level motion control, machine learning

Koray Aydin, Assistant Professor; Ph.D. Bilkent University
Research Interests: plasmonics, optical metamaterials, and semiconductor nanophotonics

Randall A. Berry, Professor; Ph.D., Massachusetts Institute of Technology
Research Interests: Wireless communication, data networking, and information theory

Larry Birnbaum, Associate Professor; Ph.D., Yale University
Research Interests: Systems, artificial intelligence, human-computer interaction, natural language processing, semantics

Fabian Bustamante, Associate Professor; Ph.D., Georgia Institute of Technology
Research Interests: Experimental systems, with a focus on operating systems, distributed and parallel computing

Arthur R. Butz, Associate Professor; Ph.D., University of Minnesota
Research Interests: Digital signal processing, median and related filtering

Yan Chen, Associate Professor; Ph.D., University of California at Berkeley
Research Interests: Computer networking and large-scale distributed systems, network measurement, diagnosis, and security, overlay and peer-to-peer systems

Alok Choudhary, Professor; Director, Center for Ultra-scale Computing and Information Security; Ph.D., University of Illinois at Urbana-Champaign
Research Interests: High-performance computing and storage, compiler and runtime systems for HPC and embedded power-aware systems, parallel data mining and databases

Oliver Cossairt, Assistant Professor, Ph.D., Columbia University
Research Interests: Optics, computer vision and computer graphics.

Peter Dinda, Professor; Ph.D., Carnegie Mellon University
Research Interests: Distributed systems, distributed interactive applications, networking, resource demand and availability prediction, performance analysis, statistical analysis and prediction

Douglas Downey, Assistant Professor; Ph.D. University of Washington
Research Interests: Natural language processing, machine learning, and artificial intelligence, using the Web to support the automatic construction of large and useful knowledge bases.

Robby Findler, Assistant Professor; Ph.D., Rice University
Research Interests: Formal methods of software design, and in particular contracts, or dynamically enforced interface specifications.

Kenneth Forbus, Professor, joint appointment with School of Education and Social Policy; Ph.D., Massachusetts Institute of Technology
Research Interests: Qualitative physics, analogical reasoning and learning, cognitive simulation, sketching as an interface modality, AI-based articulate virtual laboratories and modeling environments for education, computer game design

Randy Freeman, Associate Professor; Ph.D., University of California, Santa Barbara
Research Interests: Nonlinear control systems, robust control, adaptive control, optimal control, game theory

Matthew Grayson, Associate Professor; Ph.D. Princeton University
Research Interests: Spintronics with GaAs holes, manipulation of the valley index in AlAs nanodevices, and one-dimensional transport in novel quantum wires

Dongning Guo, Assistant Professor; Ph.D., Princeton University
Research Interests: Wireless communications, information theory, communication networks, signal processing

Abraham H. Haddad, Henry and Isabelle Dever Professor; Director, M.S. in Information Technology; Director, Council on Dynamic Systems and Control; Ph.D., Princeton University
Research Interests: Stochastic systems, modeling, estimation, detection, nonlinear filtering, singular perturbation, applications to communications and control

Kristian Hammond, Professor; Ph.D., Yale University
Research Interests: Science of case-based reasoning - understanding the role of examples and experience in reasoning; how encapsulated experience, or cases, can be used to inform planning, problem solving, and the control of action; how examples can be used in information retrieval and in communicating preferences to a machine

Nikos Hardavellas, Assistant Professor; Ph.D., Carnegie Mellon University
Research Interests: Multicore and multiprocessor architecture, memory systems, database systems, data-intensive high-performance computing

Jason D. Hartline, Associate Professor; Ph.D., University of Washington
Research Interests: Algorithmic mechanism design, algorithmic game theory, distributed algorithms, randomized algorithms, competitive analysis, data structures, machine learning theory, auction theory, microeconomics, economic theory

Lawrence J. Henschen, Professor; Ph.D., University of Illinois at Urbana-Champaign
Research Interests: Wireless sensor networks and programming; human-computer interaction (HCI) and universal access; document format systems and interoperability

Seng-Tiong Ho, Professor; Ph.D., Massachusetts Institute of Technology
Research Interests: Photonic device integration, DWDM chip technology, IV-V device modeling, nanoscale photonic device technology, micro-optics technology, organics and inorganics electro-optic modulators, quantum and non-linear optics

Michael Honig, Professor; Ph.D., University of California, Berkeley
Research Interests: Digital communications, wireless communications, networks, signal processing

Michael Horn, Assistant Professor; Ph.D., Tufts University
Joint appointment with Education & Social Policy
Research Interests: Human-computer interaction and education and innovative uses of emerging technologies in learning settings

Ian Horswill, Associate Professor; Ph.D., Massachusetts Institute of Technology
Research Interests: Autonomous agents, robotics and computer vision, cognitive architecture and situated agency and biological modeling

Russ Joseph, Assistant Professor; Ph.D., Princeton University
Research Interests: Computer architecture and power-aware computer systems including techniques for monitoring, characterizing, and optimizing performance and power consumption

Ming-Yang Kao, Professor; Ph.D., Yale University
Research Interests: Design, analysis, applications and implementation of algorithms, specific application areas include: computational biology, computational finance, and e-commerce. Specific algorithm areas include: combinatorial optimization, online computing, and parallel computing

Aggelos K. Katsaggelos, Ameritech Professor; Director, Motorola Center for Seamless Communications; Ph.D., Georgia Institute of Technology
Research Interests: Image and video recovery and compression, multimedia signal processing, computational vision, image and video restoration

Prem Kumar, AT&T Professor of Information Technology, joint appointment with the Department of Physics and Astronomy; Director, Center for Photonic Communication and Computing; Ph.D., State University of New York, Buffalo
Research Interests: Quantum and nonlinear optics, laser and atomic physics, fiber-optic communications, networks

Aleksandar Kuzmanovic, Assistant Professor, Ph.D., Rice University
Research Interests: High-speed networks, network security, multimedia communication, resource management and control in large-scale networks, network measurement and analysis

Chung-Chieh Lee, Professor; Ph.D., Princeton University
Research Interests: Digital communications, communication network performance modeling and analysis, distributed multi-sensor detection and estimation

Wei-Chung Lin, Associate Professor; Ph.D., Purdue University
Research Interests: Computer vision, pattern recognition, neural networks, and computer graphics

Chang Liu, Professor, joint appointment with Mechanical Engineering; Ph.D., California Institute of Technology
Research Interests: Sensors and sensing technology, micro and nanofabrication

Gokhan Memik, Associate Professor, Lisa Wissner-Slivka and Benjamin Slivka Chair in Computer Science; Ph.D., University of California, Los Angeles
Research Interests: Computer Architecture, embedded systems, compilers, design automation

Seda Ogrenci Memik, Associate Professor; Ph.D., University of California, Los Angeles
Research Interests: Computer-aided design for VLSI, reconfigurable computing, synthesis for programmable systems

Hooman Mohseni, Associate Professor; Ph.D., Northwestern University
Research Interests: Low-dimensional devices, quantum dots, nano-photonics, novel opto-electronic devices, novel integration methods for photonic integration circuits (PIC), advanced optical modulators

Thrasyvoulos Pappas, Professor; Ph.D., Massachusetts Institute of Technology
Research Interests: Image processing, multi-dimensional signal processing

Bryan Pardo, Assistant Professor; Ph.D., University of Michigan
Research Interests: Application of machine learning, probabilistic natural language processing, computer music, and search techniques to auditory user interfaces for HCI

Morteza Rahimi, Professor; Ph.D.
Formerly Chief Technology Officer for NU
Research Interests: fiber-optic networks

Manijeh Razeghi, Walter P. Murphy Professor; Director, Center for Quantum Devices; Ph.D. and ES-Science Doctorate, University of Paris
Research Interests: Compound semiconductor science and technology; theory, epitaxy, characterization, modeling and fabrication of quantum structures and devices operating from ultraviolet (200 nm) up to terahertz (100 micron)

Christopher Riesbeck, Associate Professor; Ph.D., Stanford University
Research Interests: Educational change through the development of tools for authoring and delivering interactive learning scenarios, and tools for asynchronous efficient high-quality mentoring

Alan V. Sahakian, Professor; Department Chair; Joint appointment with the Department of Biomedical Engineering; Ph.D., University of Wisconsin
Research Interests: Instrumentation, signal and image processing for medical and aerospace applications, automatic detection and treatment of atrial cardiac arrhythmias by implanted devices

Peter Scheuermann, Professor; Ph.D., State University of New York, Stony Brook
Research Interests: Physical database design, pictorial databases, parallel I/O systems, parallel algorithms for data-intensive applications, distributed database systems

Selim M. Shahriar, Associate Professor; Ph.D., Massachusetts Institute of Technology
Research Interests: Applications of optically induced spin transitions, nanolithography, optical data storage, and optical phase conjugation

Allen Taflove, Professor; Ph.D., Northwestern University
Research Interests: Theory and applications of computational electrodynamics, especially finite-difference time-domain (FDTD) solutions of Maxwell's equations

Goce Trajcevski, Assistant Chair, Lecturer; Ph.D., University of Illinois at Chicago
Research Interests: Mobile Data Management and Moving Objects Databases (MOD), Data Management in Sensor Networks, and Reactive Behavior in Dynamic and Distributed Environments

Jack Tumblin, Associate Professor; Ph.D., Georgia Institute of Technology
Research Interests: Human visual perception of intensity, movement, form and color; computer graphics; visual appearance; surface modeling; computational geometry; image-based rendering; image processing; and computer vision

Uri Wilensky, Associate Professor, joint appointment with the School of Education and Social Policy; Ph.D., Massachusetts Institute of Technology
Research Interests: Multi-agent modeling, modeling and simulation, networked simulation environments, parallel algorithms

Chi-haur Wu, Associate Professor; Ph.D., Purdue University
Research Interests: Robotics, CAD/CAM, industrial control applications, automated manufacturing, neural networks, computer graphics and images, automated medical instrumentations, surgical robot systems

Ying Wu, Associate Professor; Ph.D., University of Illinois, Urbana-Champaign
Research Interests: Computer vision and graphics, image and video processing, vision-based human-computer interaction, machine learning and pattern recognition, multimedia, multimodal human-computer interactions, virtual environments, robotics

Horace P. Yuen, Professor, joint appointment with the Department of Physics and Astronomy; Ph.D., Massachusetts Institute of Technology
Research Interests: Optical communication, theoretical quantum optics, measurement theory, physical cryptography

Haoqi Zhang, Assistant Professor, EECS & Segal Design Institute (Appointment begins September 2013)
Ph.D., Harvard University
Research Interests: Advancing our ability to design and study social and economic systems on the Web to promote desired participant behaviors and outcomes

Hai Zhou, Associate Professor; Ph.D., University of Texas at Austin
Research Interests: VLSI design automation including physical design, logic synthesis, and formal verification

Research Faculty

Ankit Agrawal, Research Assistant Professor (Choudhary); Ph.D.
Research Interests:

Joseph Altepeter, Research Assistant Professor (Kumar); Ph.D. Northwestern University
Research Interests:

Shaban Darvish, Research Assistant Professor (CQD); Ph.D.
Research Interests:

Thomas Hinrichs, Research Associate Professor (Forbus); Ph.D., Georgia Institute of Technology
Research Interests: Analogical and case-based reasoning, qualitative reasoning, performance support for engineering design

Kemi Jona, Research Associate Professor (Horswill); Ph.D. Northwestern University
Research Interests: Online learning, virtual schools, technologies to support online learning and collaboration, online laboratory science course design, corporate e-learning strategy and design

Wei-Keng Liao, Research Associate Professor (Choudhary); Ph.D., Syracuse U
Research Interests: High-performance computing systems, parallel input/output system design, implementation and evaluation of radar signal processing applications on HPC systems

Ryan McClintock, Research Assistant Professor (CQD); Ph.D., Northwestern U
Research Interests: Wide Band-Gap III-Nitride Semiconductors, Solar-Blind Photodetectors and UV Avalanche Photodiodes, Focal Plane Array Imaging Technology, Deep UV LEDs and UV Lasers

Minh Nguyen, Research Assistant Professor (CQD); PhD.
Research Interests:

Steven Slivken, Research Assistant Professor (CQD;) Ph.D., Northwestern U
Research Interests: Growth and fabrication of III-V semiconductors for use in optoelectronic devices, including quantum cascade lasers, QWIPs, type-II InAs/GaSb lasers/detectors, and quantum dot lasers/detectors

Zahra Vashaei Research Assistants Professor (CQD); Ph.D.,
Research Interests:

Adjunct Faculty

Geraldo Barbosa, Adjunct Professor (Kumar)

Gail Brown, Adjunct Professor (CQD); U.S. Air Force Research Laboratory

Robert Dick, Adjunct Professor (EECS Chair); University of Michigan; Ph.D., Princeton University

Mari Grechanik, Adjunct Professor (Findler)

John Hubbs, Adjunct Professor (CQD)

Scott Klasky, Adjunct Professor (Choudhary); Ph.D.

Patrick Kung, Adjunct Professor (CQD);

Sven Leyffer, Adjunct Professor (Kao); Argonne National Laboratory

Joel Mambretti, Adjunct Professor (McCormick); Director, International Center for Advanced Internet Research

Antoni Rogalski, Adjunct Professor (Razeghi); Professor, Institute of Physics, Military University of Technology

Steven Swiryn, Adjunct Professor (Sahakian); Professor, Division of Cardiology, Northwestern University

Rajeev Thakur, Adjunct Associate Professor (Choudhary); Argonne National Laboratory;

Sotirios Tsaftaris, Adjunct Professor (Katsaggelos); Ph.D.

Todd Warren, Adjunct Professor (Choudhary); Ph.D.

Faculty with Courtesy Appointments

Alvin Bayliss, Professor; Engineering Sciences and Applied Math

Robert Chang, Professor; Materials Sciences and Engineering

Darren Gergle, Assistant Professor; School of Communication

John Ketterson, Fayerweather Professor; Physics and Astronomy

Andrew Larson, Assistant Professor; Department of Radiology

Jorge Nocedal, Professor; Director, Computational Science Institute

Andrew Ortony, Professor; Education and Social Policy & Psychology Departments

Rakesh Vohra, J. L. and Helen Kellogg Professor of Managerial Economics and Decision Sciences; Kellogg School of Management

Bruce Wessels, Professor; Materials Science and Engineering

Emeritus Faculty

James A. Aagaard, Professor Emeritus; Ph.D., Northwestern University

Morris E. Brodwin, Professor Emeritus; Ph.D., Johns Hopkins University

Max Epstein, Professor Emeritus

Gilbert K. Krulee, Professor Emeritus; Ph.D., Massachusetts Institute of Technology

Gordon J. Murphy, Professor Emeritus; Ph.D., University of Minnesota

Donald Norman, Professor Emeritus; Ph.D.

Martin Plonus, Professor Emeritus; Ph.D.

James E. Van Ness, Professor Emeritus; Ph.D., Northwestern University

9. Laboratory and Computer Facilities

The EECS Department has well-equipped instruction and research laboratories for electronic circuits, digital circuits, solid-state electronics, biomedical electronics, communications, microwave techniques, real-time control systems, holography, fiber-optics, coherent light optics, digital systems design, computer systems (including distributed and parallel systems), security, networking, computer graphics, artificial intelligence, computer vision, and robotics.

We maintain several state-of the-art Undergraduate Teaching Labs equipped with Agilent and other high end lab equipment. A typical setup consists of an Oscilloscope, Data Acquisition Switch, Function Generator and Triple Power Supply. These, plus many special test setups that are designed here. This equipment is connected through a GPIB bus to a PC. Instruments are controlled, data collected and results printed in a very coherent manner. Our setups are very flexible so changes and updates are made easily by our faculty. We also maintain a wide range of electronic parts for student projects. Our “Introduction to Computer Engineering” lab is portable. Students are assigned a tool kit that they take with them to “breadboard” their assignments and later meet with their TA’s.

In addition, the Department has excellent computing facilities, with most of its computers upgraded in the last three years, and all of its computers linked to Northwestern's ever-evolving high-speed backbone network connection to the Internet.

The Labs

The Wilkinson Lab includes 23 modern dual-boot PCs running the latest Red Hat Enterprise Linux operating system and Windows 7 Enterprise. Each computer has a powerful graphics card attached to at least one large LCD monitor. Virtualization enables us to run 23 instances of Ubuntu Linux for the popular Network Security class taught by Professor Chen. A wide variety of graphics, CAD, circuit design & simulation, database, and other software packages are available for these machines. (Tech M-

The TLab ("Teaching Lab") is similarly configured, but consists of 18 machines and is located in a different room (Ford-Tech 2nd Floor Skybridge). Each lab has its own modern laser printer.

The Center for Ultra-scale Computing and Information Security (CUCIS) has several Sun Solaris and Red Hat Enterprise Linux workstations, 8-node Dell rack servers running as a cluster for parallel I/O software development purpose; 17-node Dell rack servers running as a cluster for social network software development purpose; 5-node Dell XPS desktops equipped with NVIDIA graphic cards for CUDA programming; 4-node Dell desktops for database software development; 10-node (2 rack servers + 8 desktops) for social network software development.

The Ford Motor Company Engineering Design Center is our state-of-the-art teaching facility. New designs will come to life at the center. Presented with real-world problems from clients, students can work their ideas out in the CADD (computer-aided drafting and design) lab and rapid prototyping areas located on the sub-basement level, and then move those plans up to the basement-level “factory floor.” The large flexible, barrier-free workspace
with its concrete floor is where designs actually get built. Students can use the design prototyping lab and fabrication facilities, which include machinery such as lathes, milling machines and large saws, to build design projects both large and small.

The Ford building also features a vehicle testing area, a mechatronics lab for building circuit boards, a 60-seat classroom, a conference room, research labs, group study rooms, project display areas and a student commons area. Faculty and graduate students from the department of Electrical Engineering and Computer Science have offices on the second and third floors, and labs in the sub-basement.

The Prescience Lab has a range of equipment to support research in operating systems, high performance computing, networking, and sensor networks.

The VLab ("Virtual Lab") consists of 11 dual Xeon 64-bit server computers, 2 TB of storage, and a private gigabit network. Students and faculty can create their own virtual computers on this hardware. Their virtual computers can run any operating system and they have root access.

The Ford Motor Company Engineering Design Center is our new state-of-the-art teaching facility. New designs will come to life at the center. Presented with real-world problems from clients, students can work their ideas out in the CADD (computer-aided drafting and design) lab and rapid prototyping areas located on the sub-basement level, and then move those plans to the basement-level "factory floor." The large flexible, barrier-free workspace with its concrete floor is where designs actually get built. Students can use the design prototyping lab and fabrication facilities, which include machinery such as lathes, milling machines and large saws, to build design projects both large and small.

The Ford building also features a vehicle testing area, a Mechatronics lab for building circuit boards, a 60-seat classroom, a conference room, research labs, group study rooms, project display areas and a student commons area. Faculty and graduate students from the department of Electrical Engineering and Computer Science have offices on the second and third floors, and labs in the sub-basement.