SCS Additional Minors

This page lists Additional Majors and Minors apart from those in Artificial Intelligence (http://coursecatalog.web.cmu.edu/schools-colleges/schoolofcomputerscience/artificialintelligence/), Computational Biology (http://coursecatalog.web.cmu.edu/schools-colleges/schoolofcomputerscience/undergraduatecomputatonalbiology/), Computer Science (http://coursecatalog.web.cmu.edu/schools-colleges/schoolofcomputerscience/undergraduatecomputerscience/) , Human-Computer Interaction (http://coursecatalog.web.cmu.edu/schools-colleges/schoolofcomputerscience/humancomputerinteractionprogram/) and Robotics. Select from the tabs below to view more information about each program.

Students should consult with their own academic advisor as well as the advisor for the given minor for specific double-counting rules, especially for students who are pursuing an SCS minor with a major or other minors closely related to computing. Additional help can be provided by the Associate Dean in the Computer Science Undergraduate Program office (Gates-Hillman Center, 4th Floor).

A note on SCS Concentrations: Computer Science majors are required to pursue a minor outside of SCS or a concentration within SCS. Additional majors in SCS are still allowed for Computer Science majors. Artificial Intelligence, Computational Biology, Human-Computer Interaction and Robotics majors can complete an SCS concentration if they wish, but it is not required for these degrees. Minors in SCS will not be allowed for SCS students where there is an aligned concentration. For example, an SCS student cannot minor in Machine Learning since there is a Machine Learning concentration. Consult the SCS Concentrations section for details on available SCS concentrations.

IDeATe Minors

Kelly Delaney, *Advisor*
kellydel@andrew.cmu.edu
https://ideate.cmu.edu (http://ideate.cmu.edu)

The Integrative Design, Arts and Technology (IDeATe) network offers students the opportunity to become immersed in a collaborative community of faculty and peers who share expertise, experience, and passions at the intersection of arts and technology. Students engage in active "learning by doing" in state-of-the-art maker spaces. The program addresses current and emerging real-world challenges that require disciplinary expertise coupled with multidisciplinary perspectives and collaborative integrative approaches.

The IDeATe undergraduate curriculum consists of ten areas, all of which can be taken as minors. The themes of these areas integrate knowledge in technology and the arts. Five of these minors are based in the School of Computer Science:

Animation & Special Effects Minor

Animation & Special Effects comprise a rich field of inquiry at the intersection of art, science, and technology. Students in the IDeATe *Animation & Special Effects* minor will gain experience and competency across a wide range of techniques, while learning about the diverse histories, theories, and practices of animation from renowned faculty experts and visiting artists. Coursework cultivates development of unique aesthetics and individual voice through opportunities for group critique, iteration, public screening and exhibition. Through the minor, students will: have opportunities to collaborate and connect with peers in other fields of research; develop relevant practical skills and abilities that can be applied across a variety of independent studio and industry contexts; deepen cultural sensitivities while expanding their own creative practices; and develop a compelling animation portfolio. In particular, students will gain skills and competencies in the following areas:

- Storytelling through animation
- Digital 2D and 3D animation techniques
- Expanded and experimental animation methods
- Real-time animation systems
- Motion-capture technologies
- Visual effects and procedural animation
- Rendering and compositing

Curriculum

One Computing Course - Minimum of 9 Units

		Units
15-104	Introduction to Computing for Creative Practice	10
15-110	Principles of Computing	10
15-112	Fundamentals of Programming and Computer Science	12
60-212	Intermediate Studio: Creative Coding	12

One IDeATe Portal Course - Minimum of 9 Units

		Units
60-125	IDeATe: Introduction to 3D Animation Pipeline Recommended portal course for this area	12
16-223	IDeATe Portal: Creative Kinetic Systems	10
18-090	Twisted Signals: Multimedia Processing for the Arts	10
53-322	IDeATe: Little Games/Big Stories: Indie Roleplaying Game Studio	9
60-223	IDeATe Portal: Introduction to Physical Computing	10
62-150	IDeATe Portal: Introduction to Media Synthesis and Analysis	10
82-250	Digital Realities: Introducing Immersive Technologies for Arts and Culture	9
99-361	IDeATe Portal	9

IDeATe Animation & Special Effects Courses - Minimum of 27 Units

		Units
15-463	Computational Photography	12
15-465/60-414	Animation Art and Technology	12
53-320	IDeATe Special Topics in Animation: Character Modeling	6
53-321	IDeATe Special Topics in Animation: Bipedal Rigging for Animation Production	6
53-323	IDEATE Storytelling Through Effects Animation	6
60-220	IDeATe: Technical Character Animation	10
60-333	IDeATe: Animation Rigging	10
60-335	IDeATe Special Topics in Animation: Story Development	6
60-398	Critical Studies: Social History of Animation	9
60-413	Advanced ETB: Real-Time Animation	10
60-415	Advanced ETB: Animation Studio	10
60-417	Advanced ETB: Video Art	10

Additional course options as available. Please refer to the IDeATe website for courses for the current and upcoming semester.

Double-Counting

Students may double-count up to two of their *Animation & Special Effects* minor courses toward other requirements.

Intelligent Environments Minor

Students in the *Intelligent Environments* minor are concerned with the design and realization of interactive 3D spaces, both physical and virtual. Students in this minor can explore how information and energy flow between physical, electronic, and computational spaces. By moving through space and time, we make sense of the world using our bodies. Just as we shape the environments around us, they in turn shape our experiences and senses, making us mindful of the need to develop responsible, equitable and inclusive environments. As a student in Intelligent Environments, through experimentation, hands-on learning, reflection, and documentation, you will learn:

- Analytical skills for the visualization and realization interactive spaces
- Principles of multimodal and embodied interactions
- 3-dimensional computer-aided design (CAD) for visualization, simulation, and fabrication

- The cultural context, and social and environmental implications of constructed environments

Students in this minor work in tandem with the Physical Computing and Media Design areas, which provide knowledge in key component elements of integrative intelligent environments. Accordingly, students can customize their studies by combining courses across these three concentrations with the help of their advisors.

Curriculum

One Computing Course - Minimum of 9 Units

		Units
15-104	Introduction to Computing for Creative Practice	10
15-110	Principles of Computing	10
15-112	Fundamentals of Programming and Computer Science	12
60-212	Intermediate Studio: Creative Coding	12

One IDeATe Portal Course - Minimum of 9 Units

		Units
16-223	IDeATe Portal: Creative Kinetic Systems Recommended Portal Course for this area	10
60-223	IDeATe Portal: Introduction to Physical Computing Recommended Portal Course for this area	10
18-090	Twisted Signals: Multimedia Processing for the Arts	10
53-322	IDeATe: Little Games/Big Stories: Indie Roleplaying Game Studio	9
60-125	IDeATe: Introduction to 3D Animation Pipeline	12
62-150	IDeATe Portal: Introduction to Media Synthesis and Analysis	10
82-250	Digital Realities: Introducing Immersive Technologies for Arts and Culture	9
99-361	IDeATe Portal	9

IDeATe Intelligent Environments Courses - Minimum of 27 Units

		Units
05-333	Gadgets, Sensors and Activity Recognition in HCI	12
16/54-375	IDeATe: Robotics for Creative Practice	10
16-376	IDeATe: Kinetic Fabrics	10
16-467	Introduction to Human Robot Interaction	12
18/05-540	Rapid Prototyping of Computer Systems	12
48-528	IDeATe: Responsive Mobile Environments	9
51-361	HyperSENSE: Augmenting Human Experience in Environments	9
53-558	Reality Computing Studio	12
99-362	IDeATe: Intelligent Learning Spaces	9

Additional course options as available. Please refer to the IDeATe website for courses for the current and upcoming semester.

Double-Counting

Students may double-count up to two of their *Intelligent Environments* minor courses toward other majors and minors.

Design for Learning Minor

Students in the *Design for Learning* minor, offered by the Human-Computer Interaction Institute (http://coursecatalog.web.cmu.edu/schools-colleges/schoolofcomputerscience/addlmajorsminors/hcii.cmu.edu) (HCII), combine skills to imagine, design, iterate, and evaluate effective new media systems for learning—from creating games for learning to integrating adaptive ed-tech and augmented reality experiences into diverse learning settings. In team-based collaborations, students focus on the critical design of learning platforms, products, and systems that leverage emerging technologies, learning science research, inclusive design, and data analytics to create engaging educational experiences with measurable real-world impact.

Through coursework in the minor, you will gain skills and competencies in:

- Learning design research and evaluation methods
- Concept modeling and prototyping techniques
- Learner-centered, inclusive and backward design frameworks
- Applied learning research and theory in team-based projects
- Communicating design choices and concepts to diverse stakeholders

Students in *Design for Learning* courses can bring media-making and prototyping competencies gained in other IDeATe areas (e.g. Game Design, Media Design, Physical Computing, Immersive Technologies in Arts & Humanities) to craft innovative learning experiences.

Curriculum

One Computing Course - Minimum of 9 Units

		Units
15-104	Introduction to Computing for Creative Practice	10
15-110	Principles of Computing	10
15-112	Fundamentals of Programming and Computer Science	12
60-212	Intermediate Studio: Creative Coding	12

One IDeATe Portal Course - Minimum of 9 Units

		Units
62-150	IDeATe Portal: Introduction to Media Synthesis and Analysis Recommended Portal Course for this area	10
99-361	IDeATe Portal Recommended Portal Course for this area	9
16-223	IDeATe Portal: Creative Kinetic Systems	10
18-090	Twisted Signals: Multimedia Processing for the Arts	10
53-322	IDeATe: Little Games/Big Stories: Indie Roleplaying Game Studio	9
60-125	IDeATe: Introduction to 3D Animation Pipeline	12
60-223	IDeATe Portal: Introduction to Physical Computing	10
82-250	Digital Realities: Introducing Immersive Technologies for Arts and Culture	9

IDeATe Design for Learning Courses - Minimum of 27 Units

05-291	Learning Media Design	12
05-292	IDeATe: Learning in Museums	12
05-321	Transformational Game Design Studio	12
05-418	Design Educational Games	12
05-432	Personalized Online Learning	12
05-738	Evidence-Based Educational Design	12
05-823	E-Learning Design Principles and Methods	12
51-486	Designing Experiences for Learning	9
79-343	Education, Democracy, and Civil Rights	9
82-288	Everyday Learning: Designing Learning Exp in Times of Unrest & Uncertainty	Var.
90-463	Policy and Leadership in Public Education	6
99-362	IDeATe: Intelligent Learning Spaces	9

Additional course options as available. Please refer to the IDeATe website for courses for the current and upcoming semester.

Double-Counting

Students may double-count up to two of their *Design for Learning* minor courses toward requirements for other majors and minors.

Physical Computing Minor

Physical computing is driven by a creative combination of arts and engineering disciplines. Our students' projects interact with their surroundings, remember information, make decisions, and generate tangible outputs like movement, sound, or light. Physical computing projects range from the tiny and plain (a blinking light on a breadboard) to the extravagant (a simulation of an alien landscape), and everything in between. They may be functional, like an assistive device for a person with disability, playful, like an interactive marble run, or exploratory, like a prototype for a future human-computer interface in a world of sentient machines.

Students gain a broad range of skills in our courses because physical computing as a field is fundamentally interdisciplinary: our projects combine software, electronics, and physical fabrication. Students in the *Physical Computing* minor learn how to:

- Write low-level software to computationally define a project's behavior, usually using C or Python
- Fabricate projects using techniques borrowed from various crafts and disciplines, such as making simple assemblies with paper and tape; woodworking for larger or more robust projects; textile/fabric integrations; and creating powered mechanical linkages using motors/gears/belts/bearings/etc.
- Design, test, assemble, and debug electronic circuits to bring a project to life
- Use 3-dimensional computer-aided design (CAD) for visualization, simulation, and fabrication of all of the above
- Combine digital fabrication techniques (3D printing, laser cutting, etc.) with hand craft to iterate towards creating a final, polished product

Curriculum

One Computing Course - Minimum of 9 Units

		Units
15-104	Introduction to Computing for Creative Practice	10
15-110	Principles of Computing	10
15-112	Fundamentals of Programming and Computer Science	12
60-212	Intermediate Studio: Creative Coding	12

One IDeATe Portal Course - Minimum of 9 Units

		Units
16-223	IDeATe Portal: Creative Kinetic Systems Recommended Portal Course for this area	10
60-223	IDeATe Portal: Introduction to Physical Computing Recommended Portal Course for this area	10
18-090	Twisted Signals: Multimedia Processing for the Arts	10
53-322	IDeATe: Little Games/Big Stories: Indie Roleplaying Game Studio	9
60-125	IDeATe: Introduction to 3D Animation Pipeline	12
62-150	IDeATe Portal: Introduction to Media Synthesis and Analysis	10
82-250	Digital Realities: Introducing Immersive Technologies for Arts and Culture	9
99-361	IDeATe Portal	9

IDeATe Physical Computing Courses - Minimum of 27 Units

		Units
05-333	Gadgets, Sensors and Activity Recognition in HCI	12
05/18-540	Rapid Prototyping of Computer Systems	12
15-294	Special Topic: Rapid Prototyping Technologies	5
15-394	Intermediate Rapid Prototyping	5
16/54-375	IDeATe: Robotics for Creative Practice	10
16-376	IDeATe: Kinetic Fabrics	10
16-480	IDeATe: Creative Soft Robotics	10
18/05-540	Rapid Prototyping of Computer Systems	12
18-578	Mechatronic Design	12
24-672	Special Topics in DIY Design and Fabrication	12
39-245	Rapid Prototype Design	9
48-528	IDeATe: Responsive Mobile Environments	9
62-362	IDeATe: Electronic Logics && Creative Practice	12
62-478	IDeATe: digiTOOL	9

Additional course options as available. Please refer to the IDeATe website for courses for the current and upcoming semester.

Double-Counting

Students may double-count up to two of their *Physical Computing* minor courses toward requirements for other majors and minors.

Soft Technologies Minor

Soft technologies is an emerging field of robotics, the arts, craft, and engineering with far-reaching commercial, research, and social implications. Individual disciplines address components of this burgeoning field, but the IDeATe *Soft Technologies* minor helps students integrate the pieces to be able to make significant contributions to this developing sphere. Through the courses in the minor, students weave together a rich set of established and experimental techniques in traditional soft materials (such as fibers and textiles) and new soft materials (such as current hybrid and dynamic materials) to design and create a variety of forms with applications ranging from novel to practical. Students explore the unique qualities that soft material technologies afford in design and interaction in relationship to environments and the human body— responsiveness, adaptivity, flexibility, sensitivity, morphing, and biomimicry. Students will engage in project-based inquiry, using research, experimentation, making, and reflection to inform their creativity and to develop critical perspectives. Students will be able to envision their own projects and develop sensitivities to the breadth and limitations of soft technologies.

Through coursework in the minor, you will gain skills and competencies in:

- Manipulating traditional soft materials (such as fibers and textiles) and new soft materials (such as current hybrid and dynamic materials).
- Constructing 3-dimensional forms from 2-dimensional planes.
- Articulating material and conceptual choices in discussions and critiques.
- Analyzing the relationships between materials, form, use, and content integral to making.
- Researching and engaging with contemporary and/or historical precedents in the field

Curriculum

One Computing Course - Minimum of 9 Units

		Units
15-104	Introduction to Computing for Creative Practice	10
15-110	Principles of Computing	10
15-112	Fundamentals of Programming and Computer Science	12
60-212	Intermediate Studio: Creative Coding	12

One IDeATe Portal Course - Minimum of 9 Units

		Units
62-150	IDeATe Portal: Introduction to Media Synthesis and Analysis Recommended Portal Course for this area	10
99-361	IDeATe Portal Recommended Portal Course for this area	9
16-223	IDeATe Portal: Creative Kinetic Systems	10
18-090	Twisted Signals: Multimedia Processing for the Arts	10
53-322	IDeATe: Little Games/Big Stories: Indie Roleplaying Game Studio	9
60-125	IDeATe: Introduction to 3D Animation Pipeline	12
60-223	IDeATe Portal: Introduction to Physical Computing	10
82-250	Digital Realities: Introducing Immersive Technologies for Arts and Culture	9

IDeATe Soft Technologies Courses - Minimum of 27 Units

		Units
09-227	The Culture of Color: Dyes, Chemistry, and Sustainability	9
15-367	Algorithmic Textiles Design	12
16-224	IDeATe: Re-Crafting Computational Thinking with Soft Technologies	6
16-376	IDeATe: Kinetic Fabrics	10
27-505	Exploration of Everyday Materials	9
54-346	Introduction to Costume Construction	6
54-486	Understanding Textiles	3

Additional course options as available. Please refer to the IDeATe website for courses for the current and upcoming semester.

Double-Counting

Students may double-count up to two of their *Soft Technologies* minor courses toward requirements for other majors and minors.

Information Security, Privacy, and Policy Minor

Lujo Bauer, Director

There is a growing demand for security and privacy experts, and increasing interest among CMU undergraduates in taking security and privacy courses. Security and privacy expertise is an asset in a variety of careers outside, not just in computer science, but also in areas that include business, management, and law. In addition, the policy side of security and privacy is becoming increasingly important and employers are interested in hiring people with an understanding of relevant policy issues, especially in the privacy and security area.

This minor is for undergraduate students across the university who are interested in policy issues related to security and privacy, including those who are planning careers in security/privacy as well as those who plan to focus their careers in other areas. The curriculum has been designed to accommodate students from any major as long as they have taken at least one introductory-level college programming course (such as 15-110 or 15-112).

After completing this minor, students will have a good understanding of how to identify potential security and privacy risks and relevant legal and policy issues; a working understanding of security topics such as cryptography, authentication, and Internet security protocols; as well as broad knowledge of several security- and privacy-related areas as they pertain to the design, development, deployment and management of technologies in a variety of practical contexts (e.g., Web, mobile, Internet of Things, social media, crypto currencies).

Admission

Students are not required to apply to enroll in this minor to start the required courses. However, students should declare their intent to complete the minor and submit a planned course of study to the minor director, and are encouraged to consult with the minor director on their elective course selection. In addition, students doing the independent study option must get approval from the minor director prior to enrolling in their independent study course. Finally, students must contact the minor director to certify their completion of the minor.

Curriculum

Students are required to take five courses to complete this minor with a minimum of 48 units.

INTRODUCTORY SECURITY COURSE

17-331	Information Security, Privacy, and Policy	12

Students who have taken 15-213 Introduction to Computer Systems may substitute 15-330 Introduction to Computer Security/18-330 Introduction to Computer Security

PRIVACY AND POLICY COURSE

17-333	Privacy Policy, Law, and Technology	9

Students may substitute 12-unit version of this course: 17-733, 19-608, or 95-818.

PRIVACY ELECTIVE

Complete a minimum of 9 units:		Units
17-334	Usable Privacy and Security (or 19-534 or 05-436)	9
17-702	Current Topics in Privacy Seminar (3-unit Mini)	3
17-731	Foundations of Privacy	12
17-735	Engineering Privacy in Software	12
17-880	Algorithms for Private Data Analysis	12
94-806	Privacy in the Digital Age	6

Crosslisted courses are also allowed.

TECHNOLOGY AND POLICY ELECTIVE

Complete a minimum of 9 units:		Units
17-200	Ethics and Policy Issues in Computing	9
19-211	Ethics and Policy Issues in Computing	9
17-562	Law of Computer Technology	9
19-101	Introduction to Engineering and Public Policy	12
19-402	Telecommunications Technology and Policy for the Internet Age	12
19-403	Policies of Wireless Systems	12
19-639	Policies of the Internet	12
84-387	Remote Systems and the Cyber Domain in Conflict	9

Crosslisted courses are also allowed.

ADDITIONAL APPROVED ELECTIVE

Students must complete an additional elective of 9 units or more. Students may choose an additional privacy elective or technology policy elective from the list above, or the one of the following security electives:

15-316	Software Foundations of Security and Privacy	9
15-356	Introduction to Cryptography	12
17-303	Cryptocurrencies, Blockchains and Applications	Var.
17-334	Usable Privacy and Security	9
18-335	Secure Software Systems	12
18-733	Applied Cryptography	
18-435	Foundations of Blockchains	12
18-334	Network Security	12

Students who have the necessary prerequisites may choose any approved elective from the SCS or ECE security and privacy undergraduate concentration. Check with the minor program director to determine which category of elective each course will fulfill.

Students should be careful to choose electives for which they have appropriate prerequisites. New elective options are expected as more courses are offered. Students may petition to count a course not on this list as an elective. Students should request permission *before* taking a course that is not on this list. Students may not count multiple electives that overlap substantially.

Optional Project: Subject to approval by the minor director, students may optionally count towards one of the elective requirements 9 units of an independent study or research project course in the security or privacy area, under the supervision of a faculty member in any department. In order to receive credit towards the minor, students must submit a brief project proposal to their project advisor and to the minor director and have it approved prior to conducting the project. Depending on the topic of the project, the minor director may approve credits counting towards privacy electives, technology policy electives, security electives, or some combination of these. Students may work individually, with other undergraduates, or as part of project teams with graduate students or research staff. Students involved in a group project must identify specific project components for which they are responsible. In addition, they must submit a final project report to their project advisor and the minor director that includes a literature review and describes the work they completed. Students working on a group project must each submit their own final report, which should also situate their contribution in the context of the larger project. Note, students are expected to work approximately 1 hour per week for each unit of project in which they are enrolled (e.g. 9 units = 9 hours/week of project work).

Double Counting: At most 2 of the courses used to fulfill the minor requirements may be counted towards any other undergraduate major or minor program. This rule does not apply to courses counted for general education requirements.

Language Technologies Minor

Carolyn P. Rose, *Chair*
cprose@cs.cmu.edu (awb@cs.cmu.edu)
www.lti.cs.cmu.edu/learn (http://www.lti.cs.cmu.edu/learn/)

Human language technologies have become an increasingly central component of computer science. Information retrieval, machine translation and speech technology are used daily by the general public, while text mining, natural language processing and language-based tutoring are common within more specialized professional or educational environments. The Language Technologies Institute prepares students for this world by offering a minor that gives you the opportunity to not only learn about language technologies, but to also apply that knowledge through a directed project.

Prerequisites

Prerequisites		Units
15-122	Principles of Imperative Computation	12
15-150	Principles of Functional Programming	12

Recommended		
21-241	Matrices and Linear Transformations	11

or 21-242	Matrix Theory	
15-259	Probability and Computing	12
or 21-325	Probability	
or 36-218	Probability Theory for Computer Scientists	

Curriculum

Core requirement:		
11-324	Human Language for Artificial Intelligence	12

Electives (choose 3):		
11-344	Machine Learning in Practice	12
11-411	Natural Language Processing	12
11-441	Machine Learning with Graphs	9
11-442	Search Engines	9
11-492	Speech Technology for Conversational AI	12
11-711	Advanced Natural Language Processing	12
11-731	Machine Translation and Sequence-to-Sequence Models	12
11-737	Multilingual Natural Language Processing.	12
11-747	Neural Networks for NLP	12
11-751	Speech Recognition and Understanding	12
11-752	Speech II: Phonetics, Prosody, Perception and Synthesis	12
11-761	Language and Statistics	12
11-776	Multimodal Affective Computing	12
80-180	Nature of Language: An Introduction to Linguistics	9
80-280	Linguistic Analysis	9

Project:	
A semester-long directed research project OR paper to provide hands-on experience and an in-depth study of a topic (in same area as a chosen elective)	12

Double Counting of Courses

Students may double count 11-324 Human Language for Artificial Intelligence and 80-180 Nature of Language: An Introduction to Linguistics toward any other major or minor.

Machine Learning Minor

Dr. Matt Gormley, *Program Director*
Laura Winter, *Program Coordinator*
ml-minor@cs.cmu.edu
www.ml.cmu.edu/academics/minor-in-machine-learning.html (http://www.ml.cmu.edu/academics/minor-in-machine-learning.html)

Machine learning and statistical methods are increasingly used in many application areas including natural language processing, speech, vision, robotics, and computational biology. The Minor in Machine Learning allows undergraduates to learn about the core principles of this field.

Prerequisites

The 3 prerequisite courses must be taken before a student applies to the Machine Learning Minor.

Prerequisites		Units
15-122	Principles of Imperative Computation	12
15-151	Mathematical Foundations for Computer Science	12
or 21-127	Concepts of Mathematics	
or 21-128	Mathematical Concepts and Proofs	
36-235	Probability and Statistical Inference I	9
or 36-218	Probability Theory for Computer Scientists	
or 36-219	Probability Theory and Random Processes	
or 36-225	Introduction to Probability Theory	
or 15-259	Probability and Computing	
or 21-325	Probability	

Core Courses

The Machine Learning Minor has 2 core courses that provide a foundation in the field.

Core Courses		
10-301	Introduction to Machine Learning	12
or 10-315	Introduction to Machine Learning (SCS Majors)	
10-403	Deep Reinforcement Learning & Control	12
or 10-405	Machine Learning with Large Datasets (Undergraduate)	
or 10-417	Intermediate Deep Learning	
or 10-418	Machine Learning for Structured Data	

Electives

The Machine Learning Minor requires at least 3 electives of at least 9 units each in Machine Learning. Students may select one of the following options to satisfy the electives requirement:

- 3 Principal courses
- 2 Principal courses + 1 Interdisciplinary course
- 2 Principal courses + 1 semester of CS Senior Honors Thesis or Senior Research
- 1 Principal course + 2 semesters of CS Senior Honors Thesis or Senior Research

Students should note that some of these elective courses (those at the 600-level and higher) are primarily aimed at graduate students, and so should make sure that they are adequately prepared for them before enrolling.

Graduate-level cross-listings of these courses can also be used for the ML Minor, if the student is adequately prepared for the more advanced version and the home department approves the student's registration.

Principal Electives

10-403	Deep Reinforcement Learning & Control	12
or 10-703	Deep Reinforcement Learning & Control	
10-405	Machine Learning with Large Datasets (Undergraduate)	12
or 10-605	Machine Learning with Large Datasets	
or 10-745	Scalability in Machine Learning	
10-414	Deep Learning Systems: Algorithms and Implementation	12
10-417	Intermediate Deep Learning	12
or 11-485	Introduction to Deep Learning	
or 10-707	Advanced Deep Learning	
10-418	Machine Learning for Structured Data	12
or 10-708	Probabilistic Graphical Models	
10-425	Introduction to Convex Optimization	12
or 10-725	Convex Optimization	
10-613	Machine Learning Ethics and Society	12
10-777	Historical Advances in Machine Learning	12
36-401	Modern Regression	9
Other courses as approved		

Note: Courses must come from separate lines. For example, if 10-417 Intermediate Deep Learning is used for the ML Minor, 11-485 Introduction to Deep Learning cannot also be used for the ML Minor.

Interdisciplinary Electives

02-510	Computational Genomics	12
03-511	Computational Molecular Biology and Genomics	9
10-335	Art and Machine Learning	12
10-737	Creative AI	Var.
11-411	Natural Language Processing	12
11-441	Machine Learning with Graphs	9
11-661	Language and Statistics	12
11-731	Machine Translation and Sequence-to-Sequence Models	12
11-751	Speech Recognition and Understanding	12
11-755	Machine Learning for Signal Processing	12
11-777	Multimodal Machine Learning	12
15-281	Artificial Intelligence: Representation and Problem Solving	12
15-386	Neural Computation	9
15-388	Practical Data Science	9
15-482	Autonomous Agents	12
16-311	Introduction to Robotics	12
16-385	Computer Vision	12
16-720	Computer Vision	12
16-745	Optimal Control and Reinforcement Learning	12
16-824	Visual Learning and Recognition	12

16-831	Introduction to Robot Learning	12
17-537	Artificial Intelligence Methods for Social Good	9
36-402	Advanced Methods for Data Analysis	9
36-462	Special Topics: Statistical Machine Learning	9
36-463	Special Topics: Multilevel and Hierarchical Models	9
36-700	Probability and Mathematical Statistics	12
Other courses as approved		

SCS Senior Honors Thesis

The SCS Senior Honors Thesis consists of 36 units of academic credit for this work. Up to 24 units (12 units each semester) may be counted towards the ML Minor. Students must consult with the Computer Science Department for information about the SCS Senior Honors Thesis. Once both student and advisor agree upon a project, the student should submit a one-page research proposal to the Machine Learning Concentration Director to confirm that the project will count for the Machine Learning Concentration.

07-599	SCS Honors Undergraduate Research Thesis	Var.

Senior Research

Senior research consists of 2 semesters of 10-500 Senior Research Project, totaling 24 units and counting as 2 electives.

The research must be a year-long senior project, supervised or co-supervised by a Machine Learning Core or Affiliated Faculty member. It is almost always conducted as two semester-long projects, and must be done in senior year. Some samples of available Machine Learning Senior Projects are available on the Machine Learning Department webpage.

Interested students should contact the faculty they wish to advise them to discuss the research project, before the semester in which research will take place. Once both student and advisor agree upon a project, the student should submit a one-page research proposal to the Machine Learning Minor Director to confirm that the project will count for the Machine Learning Minor.

The student should expect to meet with the Minor Director during both Senior Fall and Spring to discuss the project, and will present the work and submit a year-end write-up to the Minor Director at the end of Senior year.

10-500	Senior Research Project	24

Double Counting

No course in the Machine Learning Minor may be counted towards another SCS minor. Additionally, at least 3 courses (each being at least 9 units) must be used for only the Machine Learning Minor, not for any other major, minor, or concentration. (These double counting restrictions apply specifically to the Core Courses and the Electives. Prerequisites may be counted towards other majors, minors, and concentrations and do not count towards the 3 courses that must be used for only the Machine Learning Minor.)

GRADES

All courses for the Machine Learning Minor, including prerequisites, must be passed with a C or better.

ADMISSION

The Machine Learning Minor is open to undergraduate students in any major at Carnegie Mellon outside the School of Computer Science. (SCS students should instead consider the Machine Learning Concentration.) Students should apply for admission at least one semester before their expected graduation date, but are encouraged to apply as soon as they have taken the prerequisite classes for the minor. The application can be found on the Machine Learning Minor website.

Neural Computation Minor

Dr. Tai Sing Lee, *Director*
Melissa Stupka, *Administrative Coordinator*
https://www.cmu.edu/ni/academics/minor-in-neural-computation.html

Neural computation is a scientific enterprise to understand the neural basis of intelligent behaviors from a computational perspective. Study of neural computation includes, among others, decoding neural activities using statistical and machine learning techniques, and developing computational theories and neural models of perception, cognition, motor control, decision-making and learning. The neural computation minor allows students to learn about the brain from multiple perspectives, and to acquire the necessary background for graduate study in neural computation. Students enrolled in the minor will be exposed to, and hopefully participate in, the research effort in neural computation and computational neuroscience at Carnegie Mellon University.

The minor in Neural Computation is an intercollege minor jointly sponsored by the School of Computer Science, the Mellon College of Science, and the Dietrich College of Humanities and Social Sciences, and is coordinated by the Neuroscience Institute (https://www.cmu.edu/ni/) and the Center for the Neural Basis of Cognition (CNBC) (http://www.cnbc.cmu.edu/).

The Neural computation minor is open to students in any major of any college at Carnegie Mellon. It seeks to attract undergraduate students from computer science, psychology, engineering, biology, statistics, physics, and mathematics from SCS, CIT, H&SS and MCS.

The Neural Computation minor is open to students in any major of any college at Carnegie Mellon. It seeks to attract undergraduate students from computer science, psychology, engineering, biology, statistics, physics, and mathematics from SCS, CIT, Dietrich College and MCS. The primary objective of the minor is to encourage students in biology and psychology to take computer science, engineering and mathematics courses, to encourage students in computer science, engineering, statistics and physics to take courses in neuroscience and psychology, and to bring students from different disciplines together to form a community. The curriculum and course requirements are designed to maximize the participation of students from diverse academic disciplines. The program seeks to produce students with both basic computational skills and knowledge in cognitive science and neuroscience that are central to computational neuroscience.

APPLICATION

Students must apply for admission no later than November 30 of their senior years; an admission decision will usually be made within one month. Students are encouraged to apply as early as possible in their undergraduate careers so that the director of the Neural Computation minor can provide advice on their curriculum, but should contact the program director any time even after the deadline.

To apply, send email to the director of the Neural Computation minor Dr. Tai Sing Lee (tai@cnbc.cmu.edu) and copy Melissa Stupka (mstupka@andrew.cmu.edu). Include in your email:

- Full name
- Andrew ID
- Preferred email address (if different)
- Your class and College/School at Carnegie Mellon
- Semester you intend to graduate
- All (currently) declared majors and minors
- Statement of purpose (maximum 1 page) – Describes why you want to take this minor and how it fits into your career goals
- Proposed schedule of required courses for the Minor (this is your plan, NOT a commitment)
- Research projects you might be interested in

Curriculum

The Minor in Neural Computation will require a total of five courses: four courses drawn from the four core areas (A: neural computation, B: neuroscience, C: cognitive psychology, D: intelligent system analysis), one from each area, and one additional depth elective chosen from one of the core areas that is outside the student’s major. The depth elective can be replaced by a one-year research project in computational neuroscience. No more than two courses can be double counted toward the student’s major or other minors. However, courses taken for general education requirements of the student’s degree are not considered to be double counted. A course taken to satisfy one core area cannot be used to satisfy the course requirement for another core area. The following listing presents a set of current possible courses in each area. Other computational neuroscience courses are being developed at Carnegie Mellon and University of Pittsburgh that will also satisfy core area A requirement and the requirements will be updated as they come on-line. Substitution is possible but requires approval.

A. Neural Computation

		Units
15-386	Neural Computation	9
15-387	Computational Perception	9
15-883	Computational Models of Neural Systems	12
85-419	Introduction to Parallel Distributed Processing	9
86-375	Computational Perception	9
Pitt-Mathematics-1800 Introduction to Mathematical Neuroscience		9

B. Neuroscience

03-362	Cellular Neuroscience	9
03-363	Systems Neuroscience	9
03-365	Neural Correlates of Learning and Memory	9
42-630	Introduction to Neural Engineering (crosslisted with 18-690)	12
85-765	Cognitive Neuroscience	9
Pitt-Neuroscience 1000 Introduction to Neuroscience		9

C. Cognitive Psychology

85-211	Cognitive Psychology	9
85-213	Human Information Processing and Artificial Intelligence	9
85-412	Cognitive Modeling	9
85-419	Introduction to Parallel Distributed Processing	9
85-426	Learning in Humans and Machines	9
85-765	Cognitive Neuroscience	9

D. Intelligent System Analysis

10-301	Introduction to Machine Learning	12
or 10-315	Introduction to Machine Learning (SCS Majors)	
15-281	Artificial Intelligence: Representation and Problem Solving	12
15-386	Neural Computation	9
15-387	Computational Perception	9
15-494	Cognitive Robotics: The Future of Robot Toys	12
16-299	Introduction to Feedback Control Systems	12
16-311	Introduction to Robotics	12
16-385	Computer Vision	12
18-290	Signals and Systems	12
24-352	Dynamic Systems and Controls	12
36-225	Introduction to Probability Theory	9
36-401	Modern Regression	9
36-410	Introduction to Probability Modeling	9
42-631	Neural Data Analysis	12
42-632	Neural Signal Processing	12
86-375	Computational Perception	9
86-631	Neural Data Analysis	12

Prerequisites

The required courses in the above four core areas require a number of basic prerequisites: basic programming skills at the level of 15-110 Principles of Computing and basic mathematical skills at the level of 21-122 Integration and Approximation or their equivalents. Some courses in Area D require additional prerequisites. Area B Biology courses require, at minimum, 03-121 Modern Biology. Students might skip the prerequisites if they have the permission of the instructor to take the required courses. Prerequisite courses are typically taken to satisfy the students' major or other requirements. In the event that these basic skill courses are not part of the prerequisite or required courses of a student's major, one of them can potentially count toward the five required courses (e.g. the depth elective), conditional on approval by the director of the minor program.

Research Requirements (Optional)

The minor itself does not require a research project. The student however may replace the depth elective with a year-long research project. In special circumstances, a research project can also be used to replace one of the five courses, as long as (1) the project is not required by the student's major or other minor, (2) the student has taken a course in each of the four core areas (not necessarily for the purpose of satisfying this minor's requirements), and (3) has taken at least three courses in this curriculum not counted toward the student's major or other minors. Students interested in participating in the research project should contact any faculty engaged in computational neuroscience or neural computation research at Carnegie Mellon or in the University of Pittsburgh. A useful webpage that provides listing of faculty in neural computation is https://www.cmu.edu/ni/academics/pnc/pnc-training-faculty.html. The director of the minor program will be happy to discuss with students about their research interest and direct them to the appropriate faculty.

Fellowship Opportunities

The Program in Neural Computation (PNC) administered by the Center for the Neural Basis of Cognition currently provides 3-4 competitive full-year fellowships ($11,000) to Carnegie Mellon undergraduate students to carry out mentored research in neural computation. The fellowship has course requirements similar to the requirements of the minor. Students do not apply to the fellowship program directly. They have to be nominated by the faculty members who are willing to mentor them. Therefore, students interested in the full-year fellowship program should contact and discuss research opportunities with any CNBC faculty at Carnegie Mellon or University of Pittsburgh working in the area of neural computation or computational neuroscience and ask for their nomination by sending email to Dr. Tai Sing Lee, who also administers the undergraduate fellowship program at Carnegie Mellon. See www.cnbc.cmu.edu/training/undergraduate/undergraduate-research-fellowships-in-computational-neuroscience/ (http://www.cnbc.cmu.edu/training/undergraduate/undergraduate-research-fellowships-in-computational-neuroscience/) for details.

The Program in Neural Computation also offers a summer training program for undergraduate students from any U.S. undergraduate college. The students will engage in a 10-week intense mentored research and attend a series of lectures in neural computation. See www.cnbc.cmu.edu/training/undergraduate/summer-undergraduate-research-program-in-computational-neuroscience/ (http://www.cnbc.cmu.edu/training/undergraduate/summer-undergraduate-research-program-in-computational-neuroscience/) for application information.

Software Engineering Minor

Michael Hilton, *Director*
mhilton@andrew.cmu.edu
http://s3d.cmu.edu/education/undergrad (http://isri.cmu.edu/education/undergrad/)

Effectively building modern software systems at scale requires not just programming skills, but also engineering skills. These skills include the ability to interact effectively with customers to gather the requirements for a system in a precise way; to develop a design that resolves competing quality attributes; to make tradeoffs among schedule, cost, features, and quality to maximize value to stakeholders; to work effectively with other engineers; and to assure the quality of the delivered software system. We hear regularly from industry that these skills are crucial to them, and that they are interested in students with a strong software engineering background.

The software engineering minor is designed to teach the fundamental tools, techniques, and processes of software engineering. Through internships and a mentored project experience, students gain an understanding of the issues of scale and complexity that motivate software engineering tools and techniques. The core curriculum includes material both on engineering the software product and on the process, teamwork, and management skills that are essential to successful engineering. Graduates of the program should have the technical, process, and teamwork skills to be immediately productive in a mature engineering organization.

Admission

The Software Engineering Minor is open to non-SCS undergraduate students in any major in the university. We encourage students to submit applications no later than 3 days before the beginning of Spring and Fall course registration, so that subsequent decisions can help students plan their subsequent course schedule effectively. However, students may petition the Director for admission outside this general schedule.

To apply, send the director an email. Include in your email:

- Full name
- Andrew ID
- Preferred email address (if different)
- Semester you intend to graduate
- QPA
- All (currently) declared majors and minors, or home college if no major declared
- Statement of purpose (maximum 1 page) - Describes why you want to take this minor and how it fits into your career goals
- Proposed schedule of required courses and internship (this is your plan, NOT a commitment)

Prerequisite

17-214	Principles of Software Construction: Objects, Design, and Concurrency	12
or 15-214	Principles of Software Construction: Objects, Design, and Concurrency	

Core Course Requirements

Complete both of the following courses.

17-313	Foundations of Software Engineering	12
or 15-313	Foundations of Software Engineering	
17-413	Software Engineering Practicum	12
or 15-413	SEE 17-413 Software Engineering Practicum	

Electives

The minor requires three elective courses, one selected from each of the following categories:

1. One domain-independent course focused on technical software engineering material (min. 9 units):

Must complete at least 9 units, may comprise one 9-12 unit course or multiple minis

15-414	Bug Catching: Automated Program Verification	9
17-355	Program Analysis	12
17-356	Software Engineering for Startups	12
17-480	API Design and Implementation	12
17-653	Managing Software Development (Prerequisite 17-413 or internship)	6
17-614	Formal Methods ** Mini pair with 17-624	6
17-612	Business and Marketing Strategy **Mini: pair with either 17-626 or 17-627	6
17-622	Agile Methods ** Mini pair with another min-course of your choice from this list	6
17-623	Quality Assurance ** Mini pair with 17-443/17-643	6
17-731	Foundations of Privacy	12
17-423	Designing Large-scale Software Systems	12

Crosslisted courses allowed.

Other courses may be allowed, with prior approval from the Director of the Software Engineering Program.

2. One engineering-focused course with a significant software component (min. 9 units):

At least 9 units of the following:

15-410	Operating System Design and Implementation	15
15-412	Operating System Practicum	9
17-437	Web Application Development	12
15-440	Distributed Systems	12
17-422	Building User-Focused Sensing Systems	12
15-441	Networking and the Internet	12
15-445	Database Systems	12
18-749	Building Reliable Distributed Systems	12
67-443	Mobile Application Design and Development	12

Crosslisted courses allowed

Other courses may be allowed, with prior approval from the Director of the Software Engineering Program.

3. One course that explores computer science problems in society and industry, related to existing and emerging technologies and their associated social, political, legal, business, and organizational contexts (min. 9 units):

At least 9 units of the following:

15-390	Entrepreneurship for Computer Science	9
17-200	Ethical Dilemmas and Policy Issues in Computing	9
70-311	Organizational Behavior	9
17-331	Information Security and Privacy	12
17-333	Privacy Policy, Law, and Technology	9
17-334	Usable Privacy and Security	9
19-403	Policies of Wireless Systems	12
70-471	Supply Chain Management	9
17-562	Law of Computer Technology	9
17-781	Mobile and IoT Computing Services	12
17-801	Dynamic Network Analysis	12
17-821	Computational Modeling of Complex Socio-Technical Systems	12
88-341	Organizational Communication	9

Crosslisted courses allowed.

Required Internship and Reflection Course

A software engineering internship of a minimum of 8 full-time weeks in an industrial setting is required. The student must be integrated into a team and exposed to industry pressures. The intern may work in development, management, quality assurance, or other relevant positions. The director of the SE minor program has sole discretion in approving an internship experience based on these criteria. Students should confirm that an internship position is appropriate before accepting it, but internships that fulfill the criteria will also be accepted after the fact.

- 17-415 *Software Engineering Reflection (required 6 unit course, number to be determined, to be offered Fall semester):* Each student will conduct an analysis of some personal software engineering experience, typically (but not always) based on the engineering internship above. The student will then write and edit a short paper presenting this analysis. Initial course meetings will cover the reflective, writing, and speaking process. In later meetings, each student will present his or her experience through a 30-45 minute talk, which will be evaluated for communication skills and critical reflective content. This course is limited to enrollment of 16, and students who are admitted to the minor program are given first priority.

Double Counting Rule

At most 2 of the courses used to fulfill the minor requirements may be counted towards any other major or minor program. This rule does not apply to 17-214 (a prerequisite for the minor) or courses counted for general education requirements, nor does it apply to double-counting with the SCS General Education requirements.