David J. Malan

150 Western Avenue, Allston, Massachusetts, 02134 USA
malan@harvard.edu
https://cs.harvard.edu/malan

education

Harvard University, School of Engineering and Applied Sciences **2002 – 2007**
Doctor of Philosophy (Ph.D.), Computer Science. Research in cybersecurity and digital forensics with focus on detection of patterns in large datasets. Dissertation on *Rapid Detection of Botnets through Collaborative Networks of Peers*. Advised by Michael D. Smith.

Harvard University, Graduate School of Arts and Sciences **2002 – 2004**
Master of Science (S.M.), Computer Science. Research in sensor networks for emergency medical care.

Harvard College **1995 – 1999**
Bachelor of Arts (A.B.), *cum laude*, Computer Science; 3.9 of 4.0 GPA in field. Related studies in government, micro- and macro-economics, finance, statistics and probability, multivariate calculus, and linear algebra.

research interests

Cybersecurity, digital forensics, botnets, computer science education, distance learning, collaborative learning, computer-assisted instruction.

experience

Harvard University, Faculty of Arts and Sciences, Graduate School of Education **2007 –**
Gordon McKay Professor of the Practice of Computer Science, Member of the Faculty of Education. Senior member of the Faculty of Arts and Sciences. Instructor for CS50, largest course at Harvard University with 763 students as of 2019 (+478% since 2006) and largest MOOC on edX with 3.6M+ registrants, also offered at Yale University. Manage staff of 80+. Oversee community of 150+ high schools teaching CS50 AP, College Board-endorsed adaptation of course. Instructor for CS50 for MBAs at Harvard Business School and CS50 for JDs at Harvard Law School, higher-level adaptations of course for professional audiences.

Mindset Media, LLC **2008 – 2011**
Chief Information Officer (CIO). Responsible for advertising network's scalability, security, and capacity-planning. Designed infrastructure for collection of massive datasets capable of 500M HTTP hits per day with peaks of 10K per second. Acquired by Meebo, Inc.

Diskaster **2005 – 2008**
Founder. Started company that offered professional recovery of data from hard drives and memory cards as well as forensic investigations for civil matters.

Office of the Middlesex District Attorney, Special Investigations Division **2005**
Forensic Investigator. Assisted police and prosecutors with criminal investigations. Conducted forensic analyses of seized hardware. Recovered deleted and damaged data as evidence for trials. Drafted subpoenas.

Tufts University **2002 – 2005**
Lecturer. Instructor for *Computer Science 15: Data Structures*, a second course for undergraduates. Managed staff of 10 teaching assistants.

AirClic Inc. **2000 – 2001**
Engineering Manager. First technical hire of wireless startup, chaired by American Express's Harvey Golub and backed by $290M in class-B funding. Direct report to CTO. Built software-development and product-management teams. Drafted and managed all patents.

Franklin High School **1999 – 2000**
Mathematics Teacher. Responsible for five algebra and geometry classes, honors and non-honors.

Harvard University, Division of Continuing Education **1998 –**
Lecturer. Youngest instructor in Harvard Extension School's history at time; appointed while still an undergraduate. First instructor at university to podcast an entire course in audio and video formats, free to public; podcast ranked "best educational podcast" by Wired Magazine and featured on iTunes.

community service

American Red Cross, Disaster Services **2003 – 2012**
Emergency Medical Technician (EMT). Provided emergency medical care at local events.

university service, affiliations

Executive Fellow. Harvard Business School. February 2022 – present.

Co-Chair. Computer Science Curriculum Committee. School of Engineering and Applied Sciences, Harvard University. October 2019 – present.

Administrative Board. Harvard Extension School, Harvard University. September 2019 – present.

Senior Common Room. Mather House, Harvard University. September 2018 – present.

Faculty Associate. Berkman Klein Center for Internet & Society, Harvard University. July 2016 – present.

University Director. Board of Directors, Harvard Student Agencies. February 2016 – present.

Faculty Council. Faculty of Arts and Sciences, Harvard University. January 2016 – June 2016.

Allston Classroom Design Committee. School of Engineering and Applied Sciences, Harvard University. July 2015 – October 2016.

Member of the Faculty of Education. Graduate School of Education, Harvard University. June 2014 – present.

Provost's Task Force on SEAS Concentrators. Harvard University. April 2013 – December 2013.

HarvardX Faculty Committee. Harvard University. September 2012 – March 2018.

Faculty Standing Committee on Information Technology. Faculty of Arts and Sciences, Harvard University. September 2011 – June 2022.

program committees

Associate Program Chair. 53rd ACM Technical Symposium on Computer Science Education. Providence, Rhode Island. March 2022.

26th Annual Conference on Innovation and Technology in Computer Science Education. Paderborn, Germany. June–July 2021.

Associate Program Chair. 52nd ACM Technical Symposium on Computer Science Education. March 2021.

25th Annual Conference on Innovation and Technology in Computer Science Education. Trondheim, Norway. June 2020.

Associate Program Chair. 51st ACM Technical Symposium on Computer Science Education. Portland, Oregon. March 2020.

23rd Annual Conference on Innovation and Technology in Computer Science Education. Larnaca, Cyprus. July 2018.

49th ACM Technical Symposium on Computer Science Education. Baltimore, Maryland. February 2018.

48th ACM Technical Symposium on Computer Science Education. Seattle, Washington. March 2017.

46th ACM Technical Symposium on Computer Science Education. Kansas City, Missouri. March 2015.

45th ACM Technical Symposium on Computer Science Education. Atlanta, Georgia. March 2014.

refereed publications conferences, journals

Toward an Ungraded CS50. David J. Malan. 52nd ACM Technical Symposium on Computing Science Education. March 2021.

An Open-Source, API-Based Framework for Assessing Correctness of Code in CS50. Chad Sharp, Jelle van Assema, Brian Yu, Kareem Zidane, and David J. Malan. 25th Annual Conference on Innovation and Technology in Computer Science Education. Trondheim, Norway. June 2020.

Teaching Academic Honesty in CS50. David J. Malan. 51st ACM Technical Symposium on Computing Science Education. Portland, Oregon. March 2020.

Going Over the Cliff: MOOC Dropout Behavior at Chapter Transition. Chen Chen, Gerhard Sonnert, Philip M. Sadler, Dimitar D. Sasselov, Colin Fredericks, and David J. Malan. Distance Education. February 2020.

Computational Thinking and Assignment Resubmission Predict Persistence in a Computer Science MOOC. Chen Chen, Gerhard Sonnert, Philip M. Sadler, and David J. Malan. Journal of Computer Assisted Learning. February 2020.

Foreseeing the Endgame: Who Are the Students Who Take the Final Exam at the Beginning of a MOOC? Chen Chen, Gerhard Sonnert, Philip M. Sadler, and David J. Malan. Behaviour & Information Technology. January 2020.

From Cluster to Cloud to Appliance. David J. Malan. 18th Annual ACM Conference on Innovation and Technology in Computer Science Education. Canterbury, England. July 2013.

Streamlining Grading toward Better Feedback. Tommy MacWilliam and David J. Malan. 18th Annual ACM Conference on Innovation and Technology in Computer Science Education. Canterbury, England. July 2013.

Engaging Students through Video: Integrating Assessment and Instrumentation. Tommy MacWilliam, R.J. Aquino, and David J. Malan. 18th Annual Conference of the Northeast Region of the Consortium for Computing Sciences in Colleges. Loudonville, New York. April 2013.

CS50 Sandbox: Secure Execution of Untrusted Code. David J. Malan. 44th ACM Technical Symposium on Computer Science Education. Denver, Colorado. March 2013.

Scaling Office Hours: Managing Live Q&A in Large Courses. Tommy MacWilliam and David J. Malan. 28th Annual Conference of the Eastern Region of the Consortium for Computing Sciences in Colleges. Galloway, New Jersey. November 2012.

Moving CS50 into the Cloud. David J. Malan. 15th Annual Conference of the Northeast Region of the Consortium for Computing Sciences in Colleges. Hartford, Connecticut. April 2010.

Reinventing CS50. David J. Malan. 41st Annual ACM Technical Symposium on Computer Science Education. Milwaukee, Wisconsin. March 2010.

Virtualizing Office Hours in CS 50. David J. Malan. 14th Annual ACM Conference on Innovation and Technology in Computer Science Education. Paris, France. July 2009.

Implementing Public-Key Infrastructure for Sensor Networks. David J. Malan, Matt Welsh, and Michael D. Smith. ACM Transactions on Sensor Networks. Volume 4, Issue 4. November 2008.

Scratch for Budding Computer Scientists. David J. Malan and Henry H. Leitner. 38th Annual ACM Technical Symposium on Computer Science Education. Covington, Kentucky. March 2007.

Podcasting Computer Science E-1. David J. Malan. 38th Annual ACM Technical Symposium on Computer Science Education. Covington, Kentucky. March 2007.

Advanced Forensic Format: An Open, Extensible Format for Disk Imaging. Simson L. Garfinkel, David J. Malan, Karl-Alexander Dubec, Christopher C. Stevens, and Cecile Pham. Second Annual IFIP WG 11.9 International Conference on Digital Forensics. Orlando, Florida. January 2006.

A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography. David J. Malan, Matt Welsh, and Michael D. Smith. First IEEE International Conference on Sensor and Ad hoc Communications and Networks. Santa Clara, California. October 2004.

refereed publications workshops

Grading Qualitatively with Tablet PCs in CS 50. David J. Malan. Workshop on the Impact of Pen-Based Technology on Education. Blacksburg, Virginia. October 2009.

Exploiting Temporal Consistency to Reduce False Positives in Host-Based, Collaborative Detection of Worms. David J. Malan and Michael D. Smith. ACM Workshop on Rapid Malcode. Fairfax, Virginia. November 2006.

One Big File Is Not Enough: A Critical Evaluation of the Dominant Free-Space Sanitization Technique. Simson L. Garfinkel and David J. Malan. 6th Workshop on Privacy Enhancing Technologies. Cambridge, United Kingdom. June 2006.

Host-Based Detection of Worms through Peer-to-Peer Cooperation. David J. Malan and Michael D. Smith. ACM Workshop on Recurring Malcode. Fairfax, Virginia. November 2005.

CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care. David Malan, Thaddeus Fulford-Jones, Matt Welsh, and Steve Moulton. ACM Workshop on Applications of Mobile Embedded Systems. Boston, Massachusetts. June 2004.

CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care. David Malan, Thaddeus Fulford-Jones, Matt Welsh, and Steve Moulton. International Workshop on Wearable and Implantable Body Sensor Networks. London, United Kingdom. April 2004.

unrefereed publications

Sensor Networks for Emergency Response: Challenges and Opportunities. Konrad Lorincz, David J. Malan, Thaddeus R.F. Fulford-Jones, Alan Nawoj, Antony Clavel, Victor Shnayder, Geoff Mainland, Steve Moulton, and Matt Welsh. *IEEE Pervasive Computing*. October–December, 2004.

Low-Power, Secure Routing for MICA2 Mote. Breanne Duncan and David Malan. Harvard University Technical Report TR-06-04. March 2004.

Summary Structures for XML. David Malan. Harvard University Technical Report TR-05-04. March 2004.

Crypto for Tiny Objects. David Malan. Harvard University Technical Report TR-04-04. January 2004.

undergraduate research advising

R.J. Aquino '14. *Engaging Students through Video: Integrating Assessment and Instrumentation*.

Tommy MacWilliam '13. *Engaging Students through Video: Integrating Assessment and Instrumentation. Scaling Office Hours: Efficiently Managing Live Q&A in Large Courses. Streamlining Grading toward Better Feedback*.

Horia Mihail Teodorescu '11. *A Collective Biological Processing Algorithm for ECG Signals. Swarm Filtering and Application to MRI Mammography. Image re-morphing and feature extraction with swarming algorithm*.

invited talks

Using Visual Studio Code in the Cloud with Docker Containers: Standardizing Students' Programming Environments with GitHub Codespaces. 27th Annual Conference on Innovation and Technology in Computer Science Education. Dublin, Ireland. July 2022.

Technology We Can't Live Without! (COVID-9 edition). 52nd ACM Technical Symposium on Computing Science Education. Providence, Rhode Island. March 2022.

Git Pushing the Envelope. GitHub Universe. October 2021.

Computer Science with Theatricality: CS50's Collaboration with the American Repertory Theater. Illinois Computer Science Summer Teaching Workshop. August 2021.

Higher Education Instructor Roundtable: Teach, Learn, and Connect Best Practices. Zoom Academy 2021. July 2021.

Microteaching: Semantics, Definition of a Computer, Running Times, Fractal Trees, Classes as Encapsulation, and P vs NP. 52nd ACM Technical Symposium on Computing Science Education. March 2021.

Designing and Offering Multi-Modality Courses. UPCEA Summit for Online Leadership and Administration + Roundtable. February 2021.

How to Design Environments That Optimize Blended Learning. Zoomtopia 2020. October 2020.

DNA. Nifty Assignments, 51st ACM Technical Symposium on Computing Science Education. Portland, Oregon. March 2020.

Esto es Web50. Universidad Nacional de Ingeniería. Managua, Nicaragua. November 2019.

Algorítmos. American Nicaraguan School. Managua, Nicaragua. November 2019.

¿Qué es la informática? Universidad Francisco Marroquín. Guatemala City, Guatemala. November 2019.

Education as a Service: Containerization and Orchestration of CS50 IDE. KubeCon + CloudNativeCon North America 2019. San Diego, California. November 2019.

Education as a Service: Containerization and Orchestration of CS50 IDE. Cloud Native Revolution 2019. Philadelipha, Pennsylvania. October 2019.

Faculty Insider. edX Global Forum. Boston, Massachusetts. November 2018.

Scaling CS50: Building Learning Communities around the World. Aspen Symposium. Forum for the Future of Higher Education. Aspen, Colorado. June 2018.

Sherman Talk. Belmont Hill School. Belmont, Massachusetts. April 2018.

Technology We Can't Live Without!, revisited. 49th ACM Technical Symposium on Computing Science Education. Baltimore, Maryland. February 2018.

Building a Successful Teaching-Track Faculty Career. Professional Development Workshop for Teaching-Track Faculty, Computing Research Association's Education Committee (CRA-E). Baltimore, Maryland. February 2018.

Building CS50 on GitHub. GitHub Universe 2017. San Francisco, California. October 2017.

Things We Do That You Can Do Too. Building Learning Communities Education Conference 2017. Boston, Massachusetts. July 2017.

This is CS50x.ni. Universidad Nacional de Ingeniería. Managua, Nicaragua. June 2017.

CS50 a Escala. Universidad Nacional de Ingeniería. Managua, Nicaragua. June 2017.

CS50 at Scale. University of Manchester. Manchester, United Kingdom. March 2017.

Scaling Introductory Courses Using Undergraduate Teaching Assistants. 48th ACM Technical Symposium on Computing Science Education. Seattle, Washington. March 2017.

CS50x Assessments. Assessment Deep Dive. Microsoft Corporation. Redmond, Washington. March 2017.

CS50 Communities. Building Learning Communities Education Conference 2016. Boston, Massachusetts. July 2016.

Dockerizing CS50: From Cluster to Cloud to Appliance to Container. DockerCon 2016. Seattle, Washington. June 2016.

It's Peanut Butter Jelly Time!!! 2016 National CAS Conference for Teachers of Computing. Birmingham, United Kingdom. June 2016.

From Cambridge to New Haven. The Teagle Foundation. New York, New York. February 2016.
Online Learning and MOOCS—The Next Ten Years. Harvard Higher Education Leaders Forum. Graduate School of Education, Harvard University. Cambridge, Massachusetts. January 2016.

This is CS50 (AP). Building Learning Communities Education Conference 2015. Boston, Massachusetts. July 2015.

Computer Science Principles Curricula. CSTA Annual Conference 2015. Grapevine, Texas. July 2015.

Rebooting CS50. Microsoft Research. Redmond, Washington. June 2015.

Transforming CS50. University of Rhode Island. Kingston, Rhode Island. May 2015.

Een inspiratiebron voor vernieuwing van het programmeeronderwijs. International Keynote. Nationaal Informatica Onderwijs Congres (NIOC) 2015. Enschede, Netherlands. April 2015.

Week 4, continued. Miami Dade College. Miami, Florida. April 2015.

Virtual Computing Enviroments for Instruction in Higher Education: A Show Case of Successful Implementations and Uses. E-Learn 2014: World Conference on E-Learning. New Orleans, Louisiana. October 2014.

Pedagogy behind CS50 and CS50x. MOOCLab, University of California, Berkeley. Berkeley, California. January 2014.

Pedagogy behind CS50. Computer Science Colloquium. School of Informatics and Computing, Indiana University. January 2014.

Pedagogy behind CS50. CSE Colloquium. University of Washington. Seattle, Washington. January 2014.

Pedagogy behind CS50. CS Department Colloquium Series. Princeton University. Princeton, New Jersey. December 2013.

This was CS50x. Lifelong Kindergarten Group. Media Lab, Massachusetts Institute of Technology. Cambridge, Massachusetts. May 2013.

This is CS50. Department Lectures. Columbia University. New York, New York. May 2013.

Experimenter Award. Experiment Fund. Cambridge, Massachusetts. April 2013.

Rediscovering the Passion, Beauty, Joy, and Awe: Making Computing Fun Again, part 6. Daniel D. Garcia, Valerie Barr, Mark Guzdial, David J. Malan. 44th ACM Technical Symposium on Computer Science Education. Denver, Colorado. March 2013.

Campus Shuttle. Nifty Assignments, 44th ACM Technical Symposium on Computer Science Education. Denver, Colorado. March 2013.

This is CS50. Computer Science Colloquium. Yale University. New Haven, Connecticut. February 2013.

This is CS50. CS Seminars. College of Engineering and Mathematical Sciences, University of Vermont. Burlington, Vermont. February 2013.

This is CS50. CS4HS. Duke University. Durham, North Carolina. October 2012.

CSI: Computer Science Investigation. Nifty Assignments, 41st ACM Technical Symposium on Computer Science Education. Milwaukee, Wisconsin. March 2010.

BMP Puzzles. Nifty Assignments, 41st ACM Technical Symposium on Computer Science Education. Milwaukee, Wisconsin. March 2010.

The New CS 50. Colloquium on Computer Science Pedagogy. Carnegie Mellon. Pittsburgh, Pennsylvania. October 2009.

Teaching Computer Science in the Cloud. 2009 USENIX Annual Technical Conference. San Diego, California. June 2009.

Scratch @ Harvard. Scratch@MIT Conference. Cambridge, Massachusetts. July 2008.

Podcasting E-1: It's All About Access. Podcast Academy at Boston University. Boston, Massachusetts. April 2006.

Toward PKI for Sensor Networks. BBN Technologies. Cambridge, Massachusetts. November 2004.

talks

Education Online, before, during, and after COVID-19. Spring Reunion. Harvard Business School. Allston, Massachusetts. June 2022.

The Digital Classroom: Teaching during the COVID era. Harvard Club of Seattle. University of Seattle. April 2022.

Hello, Visitas: CS50 and the Art of Programming. Visitas Thinks Big 2022. Harvard University. Cambridge, Massachusetts. April 2022.

The Digital Classroom: Teaching during the COVID era. Harvard Club of Sarasota. March 2022.

This is GitHub in CS50: Five Years Later. GitHub Supporter Session. 53rd ACM Technical Symposium on Computing Science Education. Providence, Rhode Island. March 2022.

A Taste of CS50. Visitas Thinks Big 2021. Harvard University. Cambridge, Massachusetts. April 2021.

Class of 1973 Zoom Cocktail Party. Harvard University. Cambridge, Massachusetts. February 2021.

Blended Learning in CS50. HILT Colleague Conversation, Office of the Vice Provost for Advances in Learning. Harvard University. March 2020.

Education as a Service: Containerization and Orchestration of CS50 IDE. ABCD Committee. Harvard University. Cambridge, Massachusetts. January 2020.

CS50's Tools for Teaching and Learning. ABCD Committee. Harvard University. Cambridge, Massachusetts. May 2019.

CS50's GitHub-Based Tools for Teaching and Learning. 50th ACM Technical Symposium on Computing Science Education. Minneapolis, Minnesota. February – March 2019.

This is CS. Visitas Thinks Big 2018. Harvard University. Cambridge, Massachusetts. April 2018.

This is CS50 (VR). Visitas Thinks Big 2017. Harvard University. Cambridge, Massachusetts. April 2017.

Future of Education in the Face of Globalization and Digitalization. 9th German American Conference. Boston, Massachusetts. April 2017.

This is GitHub in CS50. GitHub Supporter Session. 48th ACM Technical Symposium on Computing Science Education. Seattle, Washington. March 2017.

From Harvard to edX to Yale to K12. Harvard Club of Washington DC. Washington, DC. April 2016.
Abstraction. Visitas Thinks Big 2016. Harvard University. Cambridge, Massachusetts. April 2016.

From Harvard to edX to Yale to K12, this is CS50, CS50x, and CS50 AP: An Introduction to the Intellectual Enterprises of Computer Science and the Art of Programming. Venture Capital 2016. Harvard Business School. April 2016.

What Tech Skills Will Liberal Arts Graduates Need for 21st Century Careers. Office of Career Services. Harvard University. Cambridge, Massachusetts. August 2015.

CS50 for MBAs: Computer Science for Business Leaders. Spring Reunion. Harvard Business School. Allston, Massachusetts. May 2015.

This is CS50. 20th Reunion of the Class of 1995. Harvard College. Cambridge, Massachusetts. May 2015.

CS50 Cult(ure). Visitas Thinks Big. Harvard College. Cambridge, Massachusetts. April 2015.

Computational Thinking. Junior Parents Weekend. Harvard College. Cambridge, Massachusetts. February 2015.

CS50 Cult(ure). Harvard Thinks Big. Harvard College. Cambridge, Massachusetts. February 2015.

This is CS50. HUIT Town Hall. Harvard University. Cambridge, Massachusetts. November 2014.

Content Distribution and Archival in a Digital Age. Harvard IT Summit. Harvard University. Cambridge, Massachusetts. June 2014.

This is CS50. Harvard Today: A Faculty Forum. Faculty of Arts and Sciences, Harvard University. Cambridge, Massachusetts. May 2014.

Hello, Computational Thinking. Visitas Thinks Big. Harvard College. Cambridge, Massachusetts. April 2014.

The Geek Shall Inherit the Earth. Master Classes. Harvard Graduate School of Education. Cambridge, Massachusetts. February 2014.

This is CS50: The Pedagogy behind Harvard's Introduction to Computer Science. Program on Innovation in Higher Education. Academic and Professional Programs for the Americas (LASPAU), Harvard University. Cambridge, Massachusetts. November 2013.

Lessons Learned from CS50x. ABCD Committee. Harvard University. Cambridge, Massachusetts. October 2013.

How can we evaluate 40,000 students? Graduate School of Arts and Sciences, Harvard University. Cambridge, Massachusetts. May 2013.

The Art of Programming. HAA Faculty Forum. Harvard University. Cambridge, Massachusetts. May 2013.

This was CS50x v1. FAS Campaign Steering Committee. Harvard University. Cambridge, Massachusetts. May 2013.

This is CS50x. Teacher Learning and Technology (T-553). Graduate School of Education, Harvard University. Cambridge, Massachusetts. February 2013.

This is CS50x. HarvardX Town Hall. Harvard University. Cambridge, Massachusetts. February 2013.

This is CS50x. School of Engineering and Applied Sciences, Harvard University. Cambridge, Massachusetts. December 2012.

This is CS50x. Campaign Advisory Group. School of Engineering and Applied Sciences, Harvard University. Menlo Park, California. November 2012.

The Geek Shall Inherit the Earth: Computer Science in an MBA Program. Fall Reunion. Harvard Business School. Allston, Massachusetts. October 2012.

The Geek Shall Inherit the Earth. 40th Reunion of the Class of 1972. Harvard College. Cambridge, Massachusetts. September 2012.

This is CS50. Alumni Day. Harvard Business School. Allston, Massachusetts. June 2012.

The Geek Shall Inherit the Earth. Visitas Thinks Big. Harvard University. Cambridge, Massachusetts. April 2012.

The Geek Shall Inherit the Earth. Alumni Day. Graduate School of Arts and Sciences, Harvard University. Cambridge, Massachusetts. April 2012.

Innovative Teaching in the Sciences at Harvard. Program on Effective Teaching in Engineering. Academic and Professional Programs for the Americas (LASPAU), Harvard University. Cambridge, Massachusetts. March 2012.

Mobile Apps Basics. ABCD Video Multimedia Group. Harvard University. Cambridge, Massachusetts. February 2012.

This is CS50. HILT Symposium. Harvard University. Cambridge, Massachusetts. February 2012.

Security. HackHarvard Incubator. Harvard University. Cambridge, Massachusetts. January 2012.

Usando a tecnologia efetivamente para melhorar o ensino de graduação. Critical Issues and Strategies for Leaders of Modern Universities. Cambridge, Massachusetts. April 2011.

Active Learning. Conversations@FAS, Harvard University. Cambridge, Massachusetts. February 2011.

The Geek Shall Inherit the Earth. Harvard Thinks Big. Cambridge, Massachusetts. February 2010.

tutorials

Birds of a Feather Who'd Like to Share Software Together: Teaching Tools that Improve Efficiency and Outcomes. 53rd ACM Technical Symposium on Computing Science Education. Providence, Rhode Island. March 2022.

Birds of a Feather Who'd Like to Share Software Together: Teaching Tools that Improve Efficiency and Outcomes. 52nd ACM Technical Symposium on Computing Science Education. March 2021.

CS50's GitHub-Based Tools for Teaching and Learning. 52nd ACM Technical Symposium on Computing Science Education. March 2021.

Interactive Programming Environments for Teachers and Students. 50th ACM Technical Symposium on Computing Science Education. March 2021.

CS50's GitHub-Based Tools for Teaching and Learning. 51st ACM Technical Symposium on Computing Science Education. Portland, Oregon. March 2020.

Birds of a Feather Who'd Like to Share Software Together: Teaching Tools that Improve Efficiency and Outcomes. 51st ACM Technical Symposium on Computing Science Education. Portland, Oregon. March 2020.

Birds of a Feather Who'd Like to Share Software Together: Teaching Tools that Improve Efficiency and Outcomes. 50th ACM Technical Symposium on Computing Science Education. Minneapolis, Minnesota. February 2019.

Interactive Programming Environments for Teachers and Students. 50th ACM Technical Symposium on Computing Science Education. Minneapolis, Minnesota. February – March 2019.

Birds of a Feather Who'd Like to Share Software Together: Teaching Tools that Improve Efficiency and Outcomes. Building Learning Communities Education Conference 2017. Boston, Massachusetts. July 2017.

A Web-Based IDE for Teaching with Any Language. Dan Armendariz, David J. Malan, and Nikolai Onken. 48th ACM Technical Symposium on Computing Science Education. Seattle, Washington. March 2017.

A Web-Based IDE for Teaching with Any Language. Dan Armendariz, David J. Malan, and Nikolai Onken. 47th ACM Technical Symposium on Computing Science Education. Memphis, Tennessee. March 2016.

Implementing a Massive Open Online Course (MOOC). 18th Annual Conference of the Northeast Region of the Consortium for Computing Sciences in Colleges. Loudonville, New York. April 2013.

Moving Your Course into the Cloud. 41st Annual ACM Technical Symposium on Computer Science Education. Milwaukee, Wisconsin. March 2010.

Starting with Scratch (literally) in CS 1. 41st Annual ACM Technical Symposium on Computer Science Education. Milwaukee, Wisconsin. March 2010.

Replacing Real Servers with Virtual Machines Using Amazon Elastic Compute Cloud (EC2). 23rd Large Installation System Administration Conference (LISA '09). Baltimore, Maryland. November 2009.

Replacing Real Servers with Virtual Machines Using Amazon Elastic Compute Cloud (EC2) and Simple Storage Service (S3). 2009 USENIX Annual Technical Conference. San Diego, California. June 2009.

courses

Computer Science 50: Introduction to Computer Science **2007 –**
Harvard College
Introduction to the intellectual enterprises of computer science and the art of programming. This course teaches students how to think algorithmically and solve problems efficiently. Topics include abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, and web programming. Languages include C, Python, and SQL plus HTML, CSS, and JavaScript. Problem sets inspired by the arts, humanities, social sciences, and sciences. Course culminates in a final project. Designed for concentrators and non-concentrators alike, with or without prior programming experience. Two thirds of CS50 students have never taken CS before. Among the overarching goals of this course are to inspire students to explore unfamiliar waters, without fear of failure, create an intensive, shared experience, accessible to all students, and build community among students.

Computer Science 100: Software Engineering in the Arts and Humanities **2018**
Harvard College
Introduction to applications of computer science (including web technologies, visualization, and database design) to domains in the arts and humanities. Emphasis on principles of software engineering and best practices, including code reviews, source control, and testing. Languages include JavaScript and SQL. Students work in teams to design and implement solutions to problems proposed by faculty from departments across campus. Offered jointly with Yale University.

Computer Science 164: Software Engineering **2012 – 2017**
Harvard College
An introduction to principles of software engineering and best practices. Students work in teams on the design, implementation, and deployment of a term-long capstone project that solves a problem on campus. Structured as supervised independent study with weekly milestones and weekly meetings for design discussions and code reviews with an advisor.

CS50 for JDs: Computer Science for Lawyers **2019 –**
Harvard Law School
This course is a variant of Harvard College's introduction to computer science, CS50, designed especially for law students. Whereas CS50 itself takes a bottom-up approach, emphasizing mastery of low-level concepts and implementation details thereof, this course takes a top-down approach, emphasizing mastery of high-level concepts and design decisions related thereto. Ultimately, it equips students with a deeper understanding of the legal implications of technological decisions made by clients. Through a mix of technical instruction and discussion, this course empowers students to be informed contributors to technology-driven conversations. In addition, it prepares students to formulate technology-informed legal arguments and opinions. Along the way, it equips students with hands-on experience with Python and SQL, languages via which they can mine data for answers themselves, as well as HTML, CSS, and JavaScript, languages with which web and, increasingly, mobile applications are built. Topics include algorithms, cloud computing, databases, networking, privacy, programming, scalability, security, and more, with a particular emphasis on understanding how the work developers do and the technological solutions they employ can impact clients. Students emerge from this course with first-hand appreciation of how it all works and all the more confident in the factors that should guide their decision-making. This course is designed for future attorneys who expect to work closely with and advise decision-makers on legal matters that impact or intersect with technology.

CS50 for MBAs: Computer Science for Managers, 1.5 credits **2015 –**
Harvard Business School
This course is a variant of Harvard College's introduction to computer science, CS50, designed especially for MBA students. Whereas CS50 itself takes a bottom-up approach, emphasizing mastery of low-level concepts and implementation details thereof, this course takes a top-down approach, emphasizing mastery of high-level concepts and design decisions related thereto. Ultimately, this course empowers students to make technological decisions even if not technologists themselves. Topics include artificial intelligence, cloud computing, networking, privacy, scalability, security, and more, with a particular emphasis on web and mobile technologies. Students emerge from this course with first-hand appreciation of how it all works and all the more confident in the factors that should guide their decision-making. This course is designed for students who expect to be analysts, founders, product managers, and problem-solvers more generally.

CS50 for MBAs: Computer Science for Managers, 3.0 credits **2019 –**
Harvard Business School
This course is a variant of Harvard College's introduction to computer science, CS50, designed especially for MBA students who'd like to be able to read and write code without becoming software developers themselves. Ultimately, this course empowers students to make technological decisions even if not technologists themselves. Topics include artificial intelligence, cloud computing, networking, privacy, scalability, security, and more, with a particular emphasis on web and mobile technologies. Students emerge from this course with first-hand appreciation of how it all works and all the more confident in the factors that should guide their decision-making. This course also prepares students to be conversant in today's and tomorrow's technologies and self-sufficient writing programs of modest size. The course introduces Python, a popular language for general-purpose programming, as well as SQL, a specialized language for navigating large data sets. Topics include: functions, which are instructions that perform tasks; return values, which are results that functions provide; conditionals, via which programs can decide whether or not to perform some action; loops, via which programs can take action again and again; variables, via which programs can remember information; objects and methods, via which code can be modularized; exceptions, via which errors can be handled; file I/O, via which data can be stored and loaded; libraries, via which code can be re-used; and more. This course is designed for students who expect to be analysts, founders, product managers, and problem-solvers more generally, especially if planning to work closely with technical teams.

Computer Science E-1a, S-1a: Understanding Technology **2017 –**
Harvard Extension School, Harvard Summer School
This course is for students who don't (yet) consider themselves computer persons. Designed for students who work with technology every day but don't necessarily understand how it all works underneath the hood or how to solve problems when something goes wrong, this course fills in the gaps, empowering students to use and troubleshoot technology more effectively. Through lectures on hardware, the Internet, multimedia, security, programming, and web development as well as through readings on current events, this course equips students for today's technology and prepares them for tomorrow's as well.

Computer Science E-1b, S-1b: Computer Science for Business Professionals **2017 –**
Harvard Extension School, Harvard Summer School
This course is a variant of Harvard College's introduction to computer science, CS50, designed especially for business professionals. Whereas CS50 itself takes a bottom-up approach, emphasizing mastery of low-level concepts and implementation details thereof, this course takes a top-down approach, emphasizing mastery of high-level concepts and design decisions related thereto. Ultimately this course empowers students to make technological decisions even if not technologists themselves. Topics include cloud computing, networking, privacy, scalability, security, and more, with an emphasis on web and mobile technologies. Students emerge from this course with first-hand appreciation of how it all works and all the more confident in the factors that should guide their decision-making. This course is designed for managers, product managers, founders, and decision-makers more generally.

Computer Science P-14315: Introduction to Programming with Python **2021 –**
Harvard Summer School
A gentle introduction to programming using a language called Python. Learn how to read and write code as well as how to test and "debug" it. Designed for students with no prior programming experience who would like to bolster their comfort with computing in anticipation of further studies in computer science. Learn about functions, arguments, and return values (oh my!); variables and types; objects and methods; conditions and Boolean expressions; and loops. Plus exceptions, file I/O, and libraries. Hands-on opportunities for lots of practice. Exercises inspired by real-world programming problems.

Computer Science P-14300: Introduction to Web Programming **2015 – 2019**
Harvard Summer School
This course introduces students to the principles and practice of web programming. The session begins with a brief introduction to HTML and CSS and quickly transitions to server-side programming with Python and client-side scripting with JavaScript. Topics include Ajax, DOM, event handling, HTTP, MVC, relational databases, and user experience. Projects include web apps with front-end user interfaces (mobile and desktop) and back-end application programming interfaces. Students must bring a laptop running the latest version of Mac OS or Windows (or any recent version of Linux).

Computer Science E-1: Understanding Computers and the Internet **1999 – 2011**
Harvard Extension School
This course is all about understanding: understanding what's going on inside your computer when you flip on the switch, why tech support has you constantly rebooting your computer, how everything you do on the Internet can be watched by others, and how your computer can become infected with a worm just by being turned on. Designed for students who use computers and the Internet every day but don't fully understand how it all works, this course fills in the gaps. Through lectures on hardware, software, the Internet, multimedia, security, privacy, website development, programming, and more, this course "takes the hood off" of computers and the Internet so that students understand how it all works and why. Through discussions of current events, students are exposed also to the latest technologies.

Computer Science E-23a: Introduction to Game Development **2018 –**
Harvard Extension School
This course picks up where Harvard College's CS50 leaves off, focusing on the development of 2D and 3D interactive games. Students explore the design of such childhood games as Super Mario Bros., Legend of Zelda, and Portal in a quest to understand how video games themselves are implemented. Via lectures and hands-on projects, the course explores principles of 2D and 3D graphics, animation, sound, and collision detection using frameworks like Unity and LÖVE 2D, as well as languages like Lua and C#. By class's end, students will have programmed several of their own games and gained a thorough understanding of the basics of game design and development.

Computer Science E-33a: Web Programming with Python and JavaScript **2018 –**
Harvard Extension School
This course picks up where Harvard College's CS50 leaves off, diving more deeply into the design and implementation of web apps with Python, JavaScript, and SQL using frameworks like Flask, Django, and Bootstrap. Topics include database design, scalability, security, and user experience. Through hands-on projects, students learn to write and use APIs, create interactive UIs, and leverage cloud services like GitHub and Heroku. By semester's end, students emerge with knowledge and experience in principles, languages, and tools that empower them to design and deploy applications on the Internet.

Computer Science E-39b: Mobile App Development with React Native **2018**
Harvard Extension School
This course picks up where Harvard College's CS50 leaves off, transitioning from web development to mobile app development with React Native, a popular open-source framework maintained by Facebook that enables cross-platform native apps using JavaScript without Java or Swift. The course introduces students to modern JavaScript (including ES6 and ES7) as well as to JSX, a JavaScript extension. Through hands-on projects, students gain experience with React and its paradigms, app architecture, and user interfaces. The course culminates in a final project for which students implement an app entirely of their own design.

Computer Science E-80: Introduction to Artificial Intelligence with Python **2020 –**
Harvard Extension School
This course explores the concepts and algorithms at the foundation of modern artificial intelligence, diving into the ideas that give rise to technologies like game-playing engines, handwriting recognition, and machine translation. Through hands-on projects, students gain exposure to the theory behind graph search algorithms, classification, optimization, reinforcement learning, and other topics in artificial intelligence and machine learning as they incorporate them into their own Python programs. By course's end, students emerge with experience in libraries for machine learning as well as knowledge of artificial intelligence principles that enable them to design intelligent systems of their own.

Computer Science E-75: Building Dynamic Websites **2008 – 2012**
Harvard Extension School
Today's websites are increasingly dynamic. Pages are no longer static HTML files but instead generated by scripts and database calls. User interfaces are more seamless, with technologies like Ajax replacing traditional page reloads. This course teaches students how to build dynamic websites with Ajax and with Linux, Apache, MySQL, and PHP (LAMP), one of today's most popular frameworks. Students learn how to set up domain names with DNS, how to structure pages with XHTML and CSS, how to program in JavaScript and PHP, how to configure Apache and MySQL, how to design and query databases with SQL, how to use Ajax with both XML and JSON, and how to build mashups. The course explores issues of security, scalability, and cross-browser support and also discusses enterprise-level deployments of websites, including third-party hosting, virtualization, colocation in data centers, firewalling, and load-balancing.

Computer Science E-76: Building Mobile Applications **2011 – 2013**
Harvard Extension School
Today's applications are increasingly mobile. Computers are no longer confined to desks and laps but instead live in our pockets and hands. This course teaches students how to build mobile apps for Android and iOS, two of today's most popular platforms, and how to deploy them in Android Market and the App Store. Students learn to write native apps for Android using Eclipse and the Android SDK, to write native apps for iPhones, iPod touches, and iPads using Xcode and the iOS SDK, and to write web apps for both platforms.

Computer Science E-259: XML with Java, Java Servlet, and JSP **2004 – 2008**
Harvard Extension School
This course introduces XML as a key enabling technology in Java-based applications. Students learn the fundamentals of XML and its derivatives, including DTD, SVG, XML Schema, XPath, XQuery, XSL-FO, and XSLT. Students also gain experience with programmatic interfaces to XML like SAX and DOM, standard APIs like JAXP and TrAX, and industry-standard software like Ant, Tomcat, Xerces, and Xalan. The course acquaints students with J2EE, including JavaServer Pages (JSP) and Java Servlet, and also explores HTTP, SOAP, web services, and WSDL. The course's projects focus on the implementation and deployment of these technologies.

Computer Science S-1: Great Ideas in Computer Science with Java **2003 – 2010**
Harvard Summer School
This course is an introduction to the most important discoveries and intellectual paradigms in computer science, designed for students with little or no previous background. We explore problem-solving methods and algorithm development using such high-level programming languages as Java and JavaScript. Students learn how to design, code, debug, and document programs using techniques of good programming style in a Linux-based environment. This course presents an integrated view of computer systems, from switching circuits and machine language through compilers and GUI design. We examine theoretical and practical limitations related to unsolvable and intractable computational problems, and the social and ethical dilemmas presented by such issues as software unreliability and invasion of privacy.

Computer Science 15: Data Structures **2002 – 2005**
Tufts University
A second course in computer science. Data structures and algorithms are studied through major programming projects in the C++ programming language. Topics include linked lists, trees, graphs, dynamic storage allocation, and recursion.

CS50S: Introduction to Programming with Scratch **2021 –**
HarvardX
An introduction to programming using Scratch, a visual programming language via which aspiring programmers can write code by dragging and dropping graphical blocks (that resemble puzzle pieces) instead of typing out text. Used at the start of Harvard College's introductory course in computer science, CS50, Scratch was designed at MIT's Media Lab, empowering students with no prior programming experience to design their own animations, games, interactive art, and stories. Using Scratch, this course introduces students to fundamentals of programming, found not only in Scratch itself but in traditional text-based languages (like Java and Python) as well. Topics include: functions, which are instructions that perform tasks; return values, which are results that functions provide; conditions, via which programs can decide whether or not to perform some action; loops, via which programs can take action again and again; variables, via which programs can remember information; and more. Ultimately, this course prepares students for subsequent courses in programming.

CS50P: Introduction to Programming with Python **2022 –**
HarvardX
An introduction to programming using a language called Python. Learn how to read and write code as well as how to test and "debug" it. Designed for students with or without prior programming experience who'd like to learn Python specifically. Learn about functions, arguments, and return values (oh my!); variables and types; conditionals and Boolean expressions; and loops. Learn how to handle exceptions, find and fix bugs, and write unit tests; use third-party libraries; validate and extract data with regular expressions; model real-world entities with classes, objects, methods, and properties; and read and write files. Hands-on opportunities for lots of practice. Exercises inspired by real-world programming problems. No software required except for a web browser, or you can write code on your own PC or Mac.