

MASTER OF SCIENCE PROGRAM IN COMPUTER SCIENCE

The Department of Computer Science (<http://www.cs.uchicago.edu/>) at the University of Chicago offers two graduate programs in computer science:

- A Master of Science (MS in CS) program designed for students seeking to enter the field or advance their careers in computer science practice. This is the MPSC program outlined below.
- A Ph.D. program that prepares students to conduct advanced research in computer science, with career pathways in both industry and academia. For more information, please refer to the Department of Computer Science (<http://collegecatalog.uchicago.edu/graduate/departments/computerscience/>) Ph.D. program listing.

The Masters Program in Computer Science (MPCS) (<https://masters.cs.uchicago.edu>) offers a comprehensive and professionally-oriented computer science education. The program's rigorous curriculum spans theory, programming, and systems, achieving a balanced blend of foundational principles and practical applications leading to technical careers in industry. The MPCS is especially well suited for students interested in Artificial Intelligence, High Performance Computing, Application Development, and Software Engineering.

The coursework in the MPCS represents a realistic balance between CS foundational theory and applied technical courses. Core classes include Programming, Algorithms, and Systems coursework. Electives include new and innovative courses to keep up with the fast-paced world of technology including Software Engineering, Big Data, Data Analytics, Machine Learning, Generative AI, High Performance Computing, Distributed Systems, Parallel Programming, Application Development, Web Development, UI/UX, Cloud Computing, and Entrepreneurship in Technology and Design, Build, Ship.

The MPCS offers the following programs of study to accommodate students with a wide range of backgrounds and interests:

9-Course MS in Computer Science (<https://masters.cs.uchicago.edu/academics/9course-ms-computer-science/>)

The 9-Course Masters in Computer Science program provides a balance between foundations in CS and skills necessary for technology careers. The program can be completed full-time or part-time.

12-Course MS in Computer Science Specialization Program (<https://masters.cs.uchicago.edu/academics/12-course-ms-computer-science/>)

The 12-Course Masters Program is for students seeking further specialization beyond the 9-course program. This program offers advanced coursework and the opportunity for an internship to provide a hands-on, industry-driven experience. Specializations can be completed in Application Development, Artificial Intelligence, High Performance Computing and Software Engineering.

Joint MBA/MPCS Program (<https://masters.cs.uchicago.edu/academics/joint-mba-mpcs-program-with-uchicago-booth-school-of-business/>) **with the Booth School of Business**

The Joint MBA/MPCS program meets the cross-functional demands of today's leading tech companies. Technology permeates everything, and true innovation requires the ability to understand and navigate both business and technology. Our joint program with UChicago's Booth School of Business enables students to earn both an MBA and an MS in Computer Science.

Students in the MPCS earn an MS in Computer Science degree. Introductory immersion courses in programming and discrete math are available to admitted students without a CS background.

Please see our website for **admission requirements and deadlines**. To view a complete list of course offerings, please visit the MPCS Course Schedule (<https://mpcs-courses.cs.uchicago.edu>).

For inquiries or questions, please email mpcs-admissions@lists.uchicago.edu.

COMPUTER SCIENCE MASTERS COURSES

MPCS 50101. Concepts of Programming. 100 Units.

In this course students will get an introduction to the field of computer science by learning to program in Python. Students will write code each week, learning the essentials of how to solve real-world problems in an object-oriented programming language. We will learn about fundamental data structures and algorithms, professional coding practices, algorithm design, automated testing, and the fundamentals of object-oriented programming. In MPCS 50101, we provide an immersive introduction to programming for students who may not have prior experience. We will use the Python programming language to explore general-purpose computer programming and algorithmic reasoning, in an object-oriented context.

Instructor(s): TBA Terms Offered: TBD

Note(s): Open only to MPCS students

MPCS 50103. Mathematics for Computer Science: Discrete Mathematics. 100 Units.

This course is an introduction to ideas and techniques from discrete mathematics that are used in computer science. It emphasizes mathematical proof and problem solving, employed on a variety of useful and interesting examples in counting, discrete probability, graphs, and basic number theory. On completion of the course, students will be practiced in using mathematical concepts and techniques to solve problems, and in expressing mathematical notions precisely. They will be able to use ideas and techniques from discrete mathematics in subsequent courses in computer science, in particular courses in the design and analysis of algorithms, networks, numerical methods, software engineering, data analysis, and machine learning.

Terms Offered: Autumn Summer Winter

Prerequisite(s): Precalculus, especially logarithms and exponentials, is a prerequisite; calculus is not required.

High-school level familiarity with sets, functions, and relations will be assumed. There are no programming prerequisites.

Note(s): Open only to MPCS students

MPCS 51030. iOS Application Development. 100 Units.

Advances in mobile technologies are changing the way that individuals and businesses use computing devices. This course will instruct students on the fundamentals of mobile application development using Apple's iOS SDK. An introduction to the Swift programming language, including object-oriented design and the model-view-controller pattern, will be covered. Using iOS APIs and tools, such as Xcode, Interface Builder and Instruments, students will be able to create fully-featured iPod Touch, iPhone, and iPad applications. Opportunities to create applications using watchOS or tvOS are possible for the final project. User interface and application design considerations specific to mobile technologies will also be explored. The course will consist of lectures, hands-on coding exercises and discussion. Weekly programming assignments will culminate into the development of a fully functioning iOS application. As a final project, each student will design and implement an application of their choice to be presented in class. Each student will also be required to present a case study featuring an app from the Apple's App Store. The studies will include a technical decomposition of the implementation (i.e. features, functionality, design, etc.) and a market analysis (i.e. competition, pricing, positioning, etc.) for the app. These case studies are designed to encourage students to gain an appreciation for the decisions companies and developers face when entering the app market.

Terms Offered: Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 51032. Advanced iOS Application Development. 100 Units.

Advances in mobile technologies are changing the way that individuals and businesses use computing devices. This course will explore real-world issues with developing robust, high-performance iOS applications for iPhone, iPod Touch, iPad, #Watch and #TV. The course will consist of lectures, hands-on coding exercises and discussion. Weekly programming assignments will be used to create a portfolio of applications using advanced iOS frameworks. Throughout the course, students will design and develop an application as a final project. Students may opt to work in collaboration with local companies or emerging start-ups for their project. These opportunities will be discussed during the first week of class and may vary by quarter.

Terms Offered: Spring

Prerequisite(s): MPCS 51030 or instructor's consent

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51033. Backends for Mobile Applications. 100 Units.

The breakneck adoption of mobile computing as a platform has transformed how businesses and users interact with their data. The expectations of being able to access your data anywhere and anytime has become the second pillar of mobile application design and development. New models, patterns and workflows are needed to connect applications to their server based data. In addition, other considerations such as privacy, scalability and cost must be balanced to meet the demands of all application stakeholders.

Terms Offered: TBD

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046, or 51100 and MPCS 51030

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 51036. Java Programming. 100 Units.

Java Programming is part of the core curriculum and is a fast-paced first course in Java for students with some prior programming experience, though not necessarily Java or any other object-oriented language. A strong emphasis will be placed on understanding the fundamentals of object-oriented programming including; inheritance, polymorphism, composition, and more generally on applying sound principles of contemporary software engineering and tools to real-world problems. In the latter half of the course, we will cover threads, design patterns, lambdas, and streams. For their final-projects, students will develop a multi-threaded, arcade-style game. The course format is both lecture and lab. We will be using git to facilitate our learning and to manage our projects. By the end of the quarter, students will have a working knowledge of git and know how to manage both local and remote repositories. Course contents include: program control, file and basic i/o, objects,

interfaces and abstract classes, complex object relationships, exceptions and debugging, GUI programming, lambdas and streams, recursion, collections, data structures, multi-threading, and advanced topics.

Terms Offered: Autumn

Prerequisite(s): Immersion programming (MPCS 50101) or passing score on programming placement exam.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51039. Mobile Software Development. 100 Units.

This course examines software engineering skills through the lens of mobile development. Students will leave with more confidence in their ability to debug, decipher complex software systems, test their code, navigate documentation, leverage version control, and learn new programming languages. We'll exercise these skills with both the Android and the iOS framework, but the goal isn't to become fluent in a mobile stack; the goal is to practice the skills. The course also capitalizes on the unique history of mobile handsets and tablets to talk about about accessibility, data privacy, sourcing ethics, and to what degree it is our responsibility as engineers to understand and prioritize these things. Students should either have a laptop that runs a Mac OS or have access to a Mac. This class will be using XCode (which comes on the Mac) and Android Studio (downloaded from <https://developer.android.com/studio>).

Terms Offered: Spring

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51040. C Programming. 100 Units.

This is an accelerated introduction to the C (not C++) Programming Language designed for students with prior programming experience. C is in many ways the lingua franca of computing, and a broad range of programming languages and related technologies derive from the basic principles of C memory management, control flow, and abstraction. Though there are many subtleties, C is not a big language, and it is expected that students will leave the course with a relatively deep understanding of the key concepts, which will then form a solid foundation for studying higher-level technologies. At the same time, C itself remains a very practical language, particularly so in areas such as scientific programming, high-performance computing, application level library design, systems programming, network programming, multi-threaded programming, etc. Students who successfully complete the course will be well prepared for subsequent MPCS courses in these areas. The course studies both fundamental and advanced C language constructs in the abstract and reinforces them through a range of exercises in the design of basic and advanced data structures, rudimentary algorithms, and API design.

Terms Offered: Autumn

Prerequisite(s): Immersion programming (MPCS 50101) or passing score on programming placement exam.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51042. Python Programming. 100 Units.

Python is a general-purpose programming language that is used in many application areas, including web development, scientific computing, graphical user interfaces, systems programming, gaming, rapid prototyping, data mining, and more. This course provides a thorough overview of the Python 3 language with an emphasis on writing idiomatic code in Python and object-oriented design patterns and is suitable for students with some prior programming experience. We will develop an understanding of the core features of the languages and gain exposure to commonly used standard-library and third-party modules.

Terms Offered: Autumn Winter

Prerequisite(s): MPCS 50101 or passing score on the MPCS programming placement exam.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51044. C/C++ for Advanced Programmers. 100 Units.

This course covers the major features of C++ in an accelerated fashion suitable both for experienced C++ programmers and programmers who are new to C++ as described in the prerequisites below. The course teaches how to get the most out of the current C++20 language, which "feels like a new language." It also discusses how to work around in older versions of C++. A dominant theme of the course is how to use the unique features of C++ to operate at a high-level of abstraction to support powerful design idioms and improve maintainability while also achieving the kind of performance and low-level control usually associated with lower-level languages such as C and even assembler language.

Terms Offered: Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51043 or 51046 or 51100 or programming experience in any language with instructor's consent

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51045. Advanced C++ 100 Units.

In this continuation of the MPCS 51044 course, we go beyond the basics to cover the powerful and surprising techniques that make C++ a different language in the hands of an expert. If you use C++ in your daily life, you and your team will see striking benefits from understanding and using C++ at a deeper level. While the course is

focused on C++, it can also be considered in many ways a course on advanced programming techniques that can be reused in any programming language to build enterprise-scale software as we build up both conceptual and practical models of advanced programming techniques.

Terms Offered: Spring

Prerequisite(s): MPCS 51044 or instructor's consent

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51046. Intermediate Python Programming. 100 Units.

Python is a general-purpose programming language that is used in many application areas, including data science, machine learning/AI, web development, scientific computing, graphical user interfaces, systems programming, gaming, rapid prototyping, and more. This course provides a thorough overview of the Python 3 language with an emphasis on writing idiomatic code in Python and object-oriented design patterns and is suitable for students with some prior programming experience. We will develop an understanding of the core features of the languages and gain exposure to commonly used standard-library and third-party modules.

Terms Offered: Autumn

Prerequisite(s): MPCS 51011 or high passing score on the MPCS programming placement exam.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51050. OO Architecture: Patterns, Technologies, Implementations. 100 Units.

This course gives hands-on experience in architecture and design and the communication of such designs in the form of patterns. There are no formal prerequisites except solid familiarity with Java and optionally familiarity with C++. The course is designed to give students a fundamental introduction to design and architectural patterns as they are implemented in large scale system architectures currently used in industry. Students will be encouraged to explore the various implementation possibilities afforded by these patterns. Trade-offs in terms of performance, development time, maintenance impact, etc. will also be discussed. Students will gain exposure to several industry-leading tools including Apache ActiveMQ and ServiceMix. Specific Learning Objectives include: Recognize and define design and architectural patterns in current common industry use Create code implementations of these patterns Be able to discuss implementation trade-offs of certain patterns with respect to others Provide exposure to several industry-standard pattern implementations Understand and be able to implement common code refactorings

Terms Offered: Spring

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51043 or 51046 or 51100. Also basic familiarity with one object-oriented programming language, such as Java, C# or C++

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51082. Introduction to Unix Systems. 100 Units.

This course aims to introduce to the fundamental concepts, principles, and abstractions that underlie the design and architecture of Unix systems. Students will learn how a Unix system works from the hardware level all the way up to the application level. The course will also focus on teaching students develop a command of the Unix shell environment by ensuring a basic understanding of Unix commands and utilities, and networking capabilities. Students will also be able to learn about the fundamentals of systems programming in Unix. After taking this course, students will develop a more-depth understanding of Unix and be able to use this knowledge to better implement programs on a Unix operating systems such as Linux or OS X.

Instructor(s): TBA Terms Offered: Autumn

Prerequisite(s): In order to register for this class, you must have taken MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 in progress or concurrently

Note(s): All non-MPCS students must meet the course prerequisites and complete the MPCS course request form for approval to register. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51083. Cloud Computing. 100 Units.

Cloud computing is being widely adopted by enterprises of all sizes due to the low initial investment required, attractive operating costs, and elastic capacity that can best serve the highly variable demands of modern applications. Software engineers must be familiar with cloud computing technologies since many new applications they develop will be deployed "in the cloud", and existing applications will often require integration with cloud-hosted services to take advantage of new capabilities. This course provides an introduction to cloud computing with specific consideration for development of highly scalable (or so-called "web-scale") web applications that leverage cloud infrastructure and platform services (IaaS and PaaS). We will also introduce software-as-a-service from the perspective of a consuming application. The course will emphasize practical applications of cloud computing technologies, with sufficient exploration of their theoretical underpinnings to inform architectural, design, and implementation decisions. We will use commercial cloud offerings provided by Amazon Web Services to build and deploy "real" cloud-hosted applications.

Terms Offered: Spring Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51087. High Performance Computing. 100 Units.

Parallel programming is ubiquitous in both the largest compute clusters and the smallest, low-power embedded devices. Though this has been the status quo for many years, achieving optimal parallel performance can still be a challenging, multi-disciplinary effort. In this course, we will focus on compute-intensive (rather than data-intensive) parallel programming, representative of numerical applications. Computer architecture and systems will be a pervasive theme, and we will discuss how parallel APIs map to the underlying hardware. We will implement and optimize C/C++ applications on large-scale, multicore CPU and GPU compute clusters. We learn widely-used parallel programming APIs (OpenMP, CUDA, and MPI) and use them to solve problems in linear algebra, Monte Carlo simulations, discretized partial differential equations, and machine learning. The majority of coding assignments can be completed in either C or C++. Certain applications will require coding portions in pure C; however, in these cases, we will cover the requisite information for those with previous exposure to only C++. Previous or concurrent courses in systems and architecture can be helpful, but no prerequisite knowledge of systems/architectures is assumed.

Terms Offered: Winter

Prerequisite(s): MPCS 51040 or 51100 or instructor's consent

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51100. Advanced Programming. 100 Units.

Advanced Programming fulfills the MPCS Core Programming requirement, but is intended for students who are joining the program with an existing degree in Computer Science, or with substantial experience in programming. This course will be taught primarily in C, including an accelerated introduction to the C language for students who have not used C before. The course will cover advanced data structures and topics in concurrent and multicore programming not covered in the Java Programming or C Programming courses.

Terms Offered: Autumn

Prerequisite(s): For students who have taken the programming immersion course, a minimum grade of A-plus and the endorsement of the MPCS 50101 instructor will be required. For students who take the programming placement exam, they must score a "High Pass" score (the minimum score for a "High Pass" is specified at the time of the exam).

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51132. Full Stack Software Engineering. 100 Units.

Full-stack Software Engineering focuses on integrating components throughout the entire stack of a typical cloud-based client-server architecture, including: Android, Quarkus, Docker, MongoDB, AWS-API-Gateway, AWS-Lambda, AWS-LightSail, AWS-Cognito, AWS-Amplify, and AWS-SES. Students will develop projects of progressive complexity using the tools and technologies introduced in the course. For their final project, students will develop an Android client in Kotlin with authentication, as well as the corresponding server-side components which they will then deploy to their own AWS instances. In addition, students will have the option to train and integrate a machine-learning model which uses TensorFlow and a convolutional neural network for image recognition. Students will acquire a good understanding of cloud computing, as well as the differences between serverful and serverless cloud applications. Students will also learn best practices in developing full-stack applications which is an essential skill for any engineer aspiring to be a software architect. Having an Android phone on which to develop is not a prerequisite for taking this course; students can just as easily use an Android emulator which creates an emulated hardware profile on a laptop/desktop partition running the Android-linux operating system.

Terms Offered: TBD

Prerequisite(s): MPCS 51036, 51040, 51042, 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): All non-MPCS students must meet the course prerequisites and complete the MPCS course request form for approval to register. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51200. Introduction to Software Engineering. 100 Units.

Writing first-class software requires top-notch architecture, design and coding skills, but successful software project execution—from identifying the need to providing support—depends on many factors besides technical prowess. This course surveys the key practices and processes that help ensure successful projects. It provides an introduction to central activities of software engineering other than just coding, such as planning, requirements, testing and management. It balances this discussion of typical engineering activities against the development process models in which they take place -- specifically, it addresses the tension between traditional plan-driven approaches and adaptive agile techniques. By examining the underlying principles of major development models, it shows how those principles address (or fail to address) the various problems encountered by project teams. Students who complete this course will gain a solid understanding of both plan-driven and agile software development principles and how to negotiate between them in different contexts.

Terms Offered: Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51205. Topics in Software Engineering. 100 Units.

This course is an intermediate approach to applied software design and development methods for use in creating efficient, reusable, and modular software. This course is offered annually but content and focus change from year to year. Methods we investigate include: distributed systems, architectures including microservices, event-driven architecture, Hybrid Transactional/Analytical Processing; software frameworks and container-based software development; and advanced techniques including multi-threading and data design. A heavy focus is on design and creativity and what constitutes creative design. This course provides hands-on experience in the architecture and design of systems and a review of best practices for the communication of that design. Issues in the landscape of software design, including complexity, constraints, progressive discovery, and limitations in communication will be explored. In this course, students will be organized into teams and each team will be provided with a set of (partial) requirements and will be responsible for the analysis, design, design documentation, and implementation in source code of a project that constitutes a complex software system. Each team of students will work through requirements analysis, expression of design using a modeling language, and implementation, and techniques and tools will be provided in order to facilitate the delivery.

Terms Offered: Autumn

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51215. Topics in Software: Making an Impact. 100 Units.

Tech work has enormous impact; our decisions can bypass regulations, affect whole populations, and influence the environment. When we don't recognize that leverage, we push things to production with massive unconsidered repercussions. To find fulfilling roles for ourselves and have positively impactful careers, we need skills and techniques to invest in, understand, and manage the power at our fingertips. Learn how to: - Reinforce your personal leverage as a trusted teammate to achieve the roles, opportunities, and impact that you'd like to have with your career -Recognize and estimate the impact of engineering choices on people's privacy, physical safety, data ownership, and the environment -Use historical product outcomes to support an inclusive and impact-aware approach to choosing product goals, target customers, and production timelines -Address objections with fellow engineers, product managers, and executives to enable change in software products -Overcome logistical hurdles like brittle, convoluted code bases, bugs of unknown origin, and repositories that no one understands -Anticipate and mitigate the risks associated with implementing software solutions

Instructor(s): Chelsea Troy Terms Offered: Spring Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): This course requires competency in Unix and Linux. If you attended the MPCS Unix Bootcamp you covered the required material. If you did not, please review the UChicago CS Student Resource Guide here: <https://uchicago-cs.github.io/student-resource-guide/>. Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51220. Applied Software Engineering. 100 Units.

In this course, we will explore practical techniques to solving modern software challenges. There's some flexibility in the syllabus, but planned topics include: -Code quality -Managing complexity -Test-driven development -Domain-driven design -Architectural design patterns -Source control techniques for small teams with Git -Security and cryptography essentials -Continuous integration & deployment -APIs and API management

Terms Offered: Spring

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51043 or 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51230. User Interface and User Experience Design. 100 Units.

Whether you're ordering a burrito, boarding a plane, chatting with friends, booking your next workout, or reading this sentence-you're likely using software with a user interface. This course will teach you how to create useful and engaging user interfaces. We will cover different methods of approaching design problems, how to conduct research to understand users, create prototypes, discuss, present, and assess design. Learning Objectives: -Implement design processes within various contexts: personal projects, startups, and structured software development teams. -Understand the needs of users and identify meaningful problems to address in order to create positive experiences. -Create prototypes of any fidelity using contemporary design tools for web, mobile, and design systems. -Articulate design decisions and provide constructive feedback.

Terms Offered: Autumn Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51235. Advanced User Interface and User Experience Design. 100 Units.

Advanced UI UX Design builds upon the skills and concepts learned in UI UX Design. Students will create end-to-end integrated experiences (e.g. phone, watch, web, tv); high fidelity interactive prototypes with motion; custom UI pattern libraries; interfaces for vehicles, connected devices, or wearables; and take on real projects of increased complexity for on-campus clients or local institutions, requiring research and iterative prototyping. Instructor(s): TBA Terms Offered: Spring

Prerequisite(s): MPCS 51230

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51238. Design, Build, Ship. 100 Units.

Turn visionaries into builders, dreamers into doers-transform ideas into launched products and services with real users in 10 weeks. This hands-on course focuses on iterative design and rapid software development, using AI-assisted tools to accelerate the entire process. Students will work individually or in small teams to build, launch, and acquire users for a single project they will work on across the quarter. Projects can be civic, entrepreneurial, creative, educational, experimental, or social impact-oriented-all ideas are welcome. Students are not required to pursue revenue-generating concepts (but can); the primary focus is on creativity, learning, and meaningful engagement with real users.

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 OR instructor consent. Experience in at least one programming language, familiarity with Git, and terminal usage.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 51240. Product Management. 100 Units.

Product management is a cross-disciplinary endeavor that sits at the intersection of software engineering, marketing, and the user experience. Product managers are expected to create products in support of business objectives, ensuring that products deliver value to customers and are feasible to build within varying sets of constraints. In this course we will introduce the role of the product manager and demonstrate the challenges faced by product managers. We will explore approaches for managing the tension that exists between software development and product delivery using the minimum viable product and the product roadmap as critical tools.

Terms Offered: Autumn Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122. This course assumes no prior knowledge of product management concepts or specific technologies. However, you may find some of the material easier to put into perspective if you're familiar with software design patterns, or have taken one of the other software engineering courses offered in the MPCS. This course requires competency in Unix and Linux. If you attended the MPCS Unix Bootcamp you covered the required material. If you did not, please review the UChicago CS Student Resource Guide here: <https://uchicago-cs.github.io/student-resource-guide/>.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51250. Entrepreneurship in Technology. 100 Units.

Many of the most successful companies have been created by technologists, but many technologists fail to consider entrepreneurship as a viable career pathway because it is difficult to gain exposure to entrepreneurship. Students in this class will experience, firsthand, new product development based on an idea conceived of by your group. Your group will nurture your idea by clearly defining your product, obtaining market feedback, building an initial proof-of-concept, and pitching to investors. While there is no requirement that your product become a new technology venture, this class is meant to serve as a launchpad for the first three months of a startup for those interested in pursuing their ideas further. The fundamental belief, however, is that the entrepreneurial experience provided in this class can support you whether you develop new products in your large corporate enterprise or do pursue entrepreneurship in a startup of your own, and all students are encouraged to consider this course no matter your career trajectory or level of technical proficiency.

Terms Offered: Autumn

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): If an undergraduate takes this course as CMSC 29512, it may not be used for CS major or minor credit. As of academic year 2023-2024, this course will no longer be cross-listed as CMSC 29512. Non-MPCS students must receive approval from program prior to registering. Request form available online This course requires competency in Unix and Linux. If you attended the MPCS Unix Bootcamp you covered the required material. If you did not, please review the UChicago CS Student Resource Guide here: <https://uchicago-cs.github.io/student-resource-guide/>.

MPCS 51260. Human-Computer Interaction. 100 Units.

This course explores the cognitive psychology and physiology behind humans' interactions with their environment and how this translates to "good design" of computer-based systems, interfaces, and machines. Emphasis is placed on the value of intentionally observing and analyzing the reciprocal relationship between humans and the design of systems in various contexts from a designer or engineer's perspective. From a practical standpoint, the course addresses the role human-computer interaction (HCI) plays in enhancing the day-to-day human experience and the value of a product to society, while increasing the bottom line for stakeholders. Topics include perception, memory, attention, mental models, accessibility, user/interaction research methods, interface design principles, and design communication strategies. At the end of this class you will be able to: -Understand aspects of cognitive psychology and human factors engineering that apply to human-computer interaction (HCI) -Identify components and principles of "good" (and bad) interaction design -Critically analyze and evaluate human-computer interaction through empirical research methods -Design mockups, prototypes, or revisions of interactive systems by applying HCI principles -Develop strategies for communicating design, usability, and interactivity

Instructor(s): TBA Terms Offered: Spring

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51300. Compilers. 100 Units.

At a high level, students should come out of the class with an understanding of: --Parts of compiler technology that are useful in general (i.e., scanning/regexes/parsing) and algorithmic ideas that come up in other contexts (e.g., graph algorithms - dominance and coloring, for example - and lattice algorithms). --Why some aspects of optimization are hard, both in terms of algorithmic complexity and practical concerns (e.g., supporting separate compilation), and how choices made about the language semantics can make this harder or easier. --How compile time vs. expected performance gain affects the design of compilers, especially JIT compilers.

Terms Offered: Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51400. Functional Programming. 100 Units.

This course will provide an introduction to software development in the functional paradigm, with a focus on pure, statically-typed functional programming. The functional paradigm isn't a language or technology--it's an entire school of thought on what a computer program is and how to write one. It has a long history, based on a model of computation developed by the mathematician Alonzo Church in the 30s. The functional paradigm contrasts with the imperative paradigm, which most of today's existing software is written in. But although the functional paradigm has yet to go fully mainstream, the imperative world has been taking most of its bleeding-edge ideas about software design from the functional world, particularly in recent years. And to all appearances, that trend is on the rise, with expansive new purely functional software and tooling currently under development at companies like Facebook, Google, Microsoft, GitHub, and countless others. In addition, the library ecosystem for languages such as Haskell and OCaml has become sufficiently rich and extensive that an increasing proportion of today's successful software companies (including Jane Street Capital and Input-Output) have been able to develop their entire code base purely functionally.

Terms Offered: Autumn Spring

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51043 or 51046 or 51100 or CAPP 30122 or MACS 30122.

Students with an existing CS background may take this class concurrently with Core Programming with consent from the MPCS.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51410. Object Oriented Programming. 100 Units.

This course concentrates on three major themes: Software Architecture, Object Oriented Analysis and Domain-Driven Design, and Methodology. The bulk of the course will involve advanced concepts in Object-Oriented Analysis and Design and Domain-Driven Design (OOAD/DDDD). The methods we will study include Object-Oriented Analysis and Design, Domain-Driven Design, and the Unified Modeling Language (UML). While the focus of the course is on current best practices in designing object-oriented software, the general theme of the course is coming to terms with complexity in software systems and domains. This course focuses on principles, concepts, processes, methods, and best practice models that are implemented with (most) any object-oriented programming language. Code examples that illuminate key concepts in OO design will be provided in various OO languages (Java, C++, C#, Smalltalk, Python, Common Lisp Object System (CLOS)) in order to further illuminate the concepts being discussed. For example, we will see how Python, Java, C++, and CLOS, implement, say, polymorphism, with distinctive hermeneutical and pragmatic significance. However, it is the concepts that are central, not their particular language implementations. A primary focus of the course will be to come to terms with many common patterns in software design, which provide proven and repeatable templates on which to base implementations.

Terms Offered: Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 51900. Introduction to Systems Programming. 100 Units.

Introduction to Systems Programming introduces students to the foundational concepts and techniques used to build low-level software systems in Unix environments. Students will learn how programs interact directly with operating systems, hardware, and memory through topics such as processes, memory management, file I/O, signals, threads, interprocess communication, and network programming using the C programming language. The course emphasizes hands-on programming and debugging, giving students practical experience working with system calls, command-line tools, and performance-oriented software development. By the end of the course, students will have a strong understanding of how modern computer systems operate beneath high-level applications and will be prepared for more advanced courses in computer systems.

Terms Offered: TBD

MPCS 52010. Computer Architecture. 100 Units.

Computer architecture is the science and art of selecting and interconnecting hardware components to create a computer that meets functional, performance and cost goals to run software. With the recent switch from uniprocessor to multicore microprocessors, it has become more important for the programmer to understand the hardware which will run their software. For programs to run fast now, they must become parallel. What this means to the programmer varies depending if they are dealing with an imbedded system, mobile smart phones, laptops or cloud servers. This class will introduce students to the architectural knowledge they need to write high performance software for modern systems.

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 52011. Introduction to Computer Systems. 100 Units.

This course is all about constructing your own knowledge of computer systems by building a general-purpose computer system from the ground up. The objective is to integrate key ideas from algorithms, computer architecture, operating systems, compilers, and software engineering into one unified framework. Along the way, we'll explore ideas and techniques used in the design of modern hardware and software systems, and discuss major trade-offs and future trends. Throughout this journey, you'll gain lots of cross-section views of the field of computer science, from the bare-bone details of switching circuits to the high-level abstraction of object-based software design. By the end of the course, you will have written a computer game in an object-oriented programming language; compiled that program into machine language using the compiler, the virtual machine language translator, and the assembler that you wrote; and run your program on (virtual) hardware that you designed.

Terms Offered: Autumn Spring Summer Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 completed or can take concurrently.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 52015. Advanced Computer Systems. 100 Units.

This course focuses on studying modern computer systems from the point of view of a programmer, with an emphasis on topics which help you improve the performance, correctness or utility of user-level programs. As such, this is intended to be a practical, hands-on study of contemporary computer systems. We will focus on the X86-64 architecture (as implemented by Intel/AMD 64 bit processors). Topics: - Representing and Manipulating information: unsigned and two's complement representation, IEEE floating point and corresponding arithmetic. - Machine level representation of programs: x86-64 assembly, control instructions, translation of basic C control constructs (such as loops and switch statements), a study of common code security vulnerabilities (such as buffer overflows). - Processor architecture: study of a pipelined out of order processor. - Code optimization - Memory hierarchy: persistent storage(magnetic spinning disks, SSD), RAM and ROM, and caches. - Virtual Memory This is a hands on course; There will be multiple labs requiring you to program in C.

Terms Offered: Winter

Prerequisite(s): MPCS 51040 C Programming or MPCS 51100 Advanced Programming.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 52018. Advanced Computer Architecture. 100 Units.

This course focuses on computer architecture from the perspective of a high performance programmer. It is aimed at students looking to demystify the process of coding, profiling, and optimizing algorithms, with a particular focus on systems geared toward performance intensive computing: clusters, Multicore CPUs, GPUs, RISC-V, ARM, TPUs, etc. The fundamental principles of computer architecture will be covered in depth, but an equal emphasis will be given to hands-on experience in learning how measure, tune, and report performance on common HPC resources. While a key outcome of the course is the ability to write more efficient code, the more enduring goal is to educate students to "see their code" from the hardware's perspective, broadening and deepening their understanding of programming with potential benefits across a broad range of disciplines.

Topics include: -history of HPC architectures -efficient programming for cache-based serial processors -MIMD and cache coherency -efficient programming for multicore processors -SIMD and vector instructions -GPU memory and cores -TPUs, matrix engines, systolic arrays -ARM -RISC-V

Prerequisite(s): B+ or higher in MPCS 51036, 51040, 51042, 51046, 51100. Recommended MPCS 52011.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 52040. Distributed Systems. 100 Units.

This course focuses on the theory and practice of distributed systems. Modern applications and services are increasingly distributed due to growing data sizes, plateauing sequential processing power, and the enormous number of connected devices. Virtually all web, mobile, and even PC applications now rely on networked services, calling out to backend servers to perform various functions, and many individual applications are implemented as a collection of cooperating processes or services. For example, distributed systems are used in massively multiplayer online games, cloud services, e-commerce and banking systems, peer-to-peer networks, social network, self-driving cars, telecommunication systems, and distributed databases and file systems. In this course we will explore the need for distributed systems, understand characteristics of distributed systems,

investigate where distributed systems are used, review the unique challenges of distributed systems, analyze solutions for common distributed systems problems, and gain practical knowledge of the systems and algorithms for building real distributed systems.

Terms Offered: Spring

Prerequisite(s): MPSC 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPSC 52060. Parallel Programming. 100 Units.

Parallel computing is found everywhere in modern computing. Multi-core CPUs and GPUs, supercomputers, and even mobile devices such as smartphones all provide ways to efficiently utilize parallel processing on these architectures and devices. The goal of this course is to provide an introduction to the foundations of parallel programming and to consider the performance gains and trade-offs involved in implementing and designing parallel computing systems. Specifically, this course will place an emphasis on concepts related to parallel programming on multicore processors.

Terms Offered: Autumn Spring

Prerequisite(s): MPSC 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122. Familiarity with C, Java, and/or Python

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPSC 52072. GPU Programming. 100 Units.

This course provides an introduction to programming for Graphics Processing Units (GPUs), primarily focusing on both the theoretical aspects and practical implementations for programming on them. With the increasing demand for high-performance computing, understanding how to leverage GPUs is becoming essential for various fields including machine learning, scientific computing, and computer graphics. Students will learn the fundamental principles of GPU architecture, which includes GPU hardware structure, memory hierarchy, and execution model. Students will gain proficiency in writing GPU programs by using industry-standard frameworks such as CUDA and OpenCL. By the end of this course, students will know how to proficiently use GPU resources to accelerate the performance of a wide range of applications that require their computational power.

Terms Offered: Summer

Prerequisite(s): MPSC 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online at <https://mpcs-internal.cs.uchicago.edu>.

MPSC 52083. Advanced Topics in Cloud Computing. 100 Units.

Cloud infrastructure greatly simplifies the development of highly scalable applications, but operating these applications reliably and securely at scale introduces additional challenges. In this class we will explore approaches for meeting these challenges and develop an understanding of enabling technologies. The class comprises three parts. The first part focuses on preparing an application for production operations and automating deployment of the application and its runtime infrastructure. The second part introduces security considerations and adds monitoring to the runtime environment. The third part explores ways to address some of the challenges faced by applications that deal with streaming data. Additional information on topics, assignments, and grading is available at <https://www.atmpcs.org>

Prerequisite(s): B+ or above in MPSC 51083 Cloud Computing MPSC 52553 Web Development (recommended) MPSC 52040 Distributed Systems (recommended)

MPSC 52553. Web Development. 100 Units.

This course is intended to prepare students with a general programming background to work on teams producing modern web applications. Students will learn a strong foundation of core web technologies and protocols, an overview of the major design patterns in the history of web development, and a detailed introduction to the current industry standard. We will have an emphasis on learning from publicly-available documentation so that students are equipped to learn new techniques and frameworks in this rapidly-evolving field. Specifically, the course will cover content layout and styling with HTML and CSS, dynamically generating page content on the webserver, interacting with databases, interacting with remote resources using HTTP and REST, client-side interactivity with modern Javascript, and the creation of single-page applications. This course uses agile software techniques to build real, working software each week. We will work as closely as possible on how software is developed in the industry, and all work is asynchronous, open-note, and open-internet, with collaboration, encouraged.

Terms Offered: Spring Winter

Prerequisite(s): MPSC 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPSC 52555. Backends for Applications. 100 Units.

The purpose of this class is to learn how to build applications at scale, by providing you with the techniques and tools capable of providing subsecond response times to millions of users interacting with petabytes of data. In this course, we will cover both the theory and practice of building Big Data application. We will not

only learn how to use technologies such as HDFS, MapReduce, Spark, Kafka, Hive, Thrift, HBase, Zookeeper, columnar stores, etc., but also understand why Big Data applications employ such a diverse array of technologies and where each one of them fits. We will demonstrate the practice of Big Data application architecture by implementing a running Big Data web application for exploring the relationship between weather and flight performance utilizing all of the weather and flight delay information in the United States over the last decade to explore the relationship between weather and flight performance. To develop a sound understanding of the theory of Big Data, we will learn about important formulations of Big Data application architectures, such as Nathan Marz' lambda architecture, proper use of normalized and denormalized data stores within large-scale web applications, application of the CAP theorem, etc. We will also continuously keep in mind important additional topics that invariably arise in real world applications of Big Data, such as budgeting, compliance, etc.. Students are required to bring a laptop to class every

Prerequisite(s): MPCS 51036, 51040, 51042, 51046, or 51100 or CAPP 30122 or MACS 30122.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://masters.cs.uchicago.edu>.

MPCS 52560. Applied Financial Technology. 100 Units.

Applied Financial Technology (FinTech) is an applied, survey-based course into the concepts and technology underpinning financial innovation today. Themes covered include US equities and trading algorithms, bond valuation, payments APIs, and asset tokenization and are based on the instructor's active and broad experience as a FinTech entrepreneur and passion for finance and economics. Students will be expected to do some amount of coding in Python or Javascript each week. Work will be group-based throughout the quarter with the exception of an individual final. Readings also will play a key role in understanding elements of finance which are not specifically related to technology. These foundational concepts will provide the student with a solid appreciation of sound financial innovation across time and avoid some of the trappings of speculative manias that financial innovation can produce. We will have a weekly, current topics sessions where we do a quick dive into different themes that are timely and in the news. Previous discussions have included cryptocurrencies, stablecoins, inflation, interest rate changes, and more. These enable us to balance a set curriculum with some of the items that emerge unexpectedly each week and also enable us to draw on our growing knowledge of FinTech.

Terms Offered: Autumn

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 53001. Databases. 100 Units.

Students will learn database design and development and will build a simple but complete web application powered by a relational database. We start by showing how to model relational databases using the prevailing technique for conceptual modeling -- Entity-Relationship Diagrams (ERD). Concepts covered include entity sets and relationships, entity key as a unique identifier for each object in an entity set, one-one, many-one, and many-many relationships as well as translational rules from conceptual modeling (ERD) to relational table definitions. We also examine the relational model and functional dependencies and their application to the methods for improving database design: normal forms and normalization. After design and modeling, students will learn the universal language of relational databases: SQL (Structured Query Language). We start by introducing relational algebra -- the theoretical foundation of SQL. Then we examine in detail the two aspects of SQL: data definition language (DDL) and the data manipulation language (DML). Concepts covered include subqueries (correlated and uncorrelated), aggregation, various types of joins including outer joins and syntax alternatives. Students will gain significant experience with writing and reading SQL queries throughout the course in the detailed discussions in class, online homework, and the real-world individual project.

Terms Offered: Autumn Spring

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 concurrently or completed.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 53014. Big Data Application Architecture. 100 Units.

The purpose of this class is to learn how to build applications at scale, by providing you with the techniques and tools capable of providing subsecond response times to millions of users interacting with petabytes of data. In this course, we will cover both the theory and practice of building Big Data application. We will not only learn how to use technologies such as HDFS, MapReduce, Spark, Kafka, Hive, Thrift, HBase, Zookeeper, columnar stores, etc., but also understand why Big Data applications employ such a diverse array of technologies and where each one of them fits. We will demonstrate the practice of Big Data application architecture by implementing a running Big Data web application for exploring the relationship between weather and flight performance utilizing all of the weather and flight delay information in the United States over the last decade to explore the relationship between weather and flight performance. To develop a sound understanding of the theory of Big Data, we will learn about important formulations of Big Data application architectures, such as Nathan Marz' lambda architecture, proper use of normalized and denormalized data stores within large-scale web applications, application of the CAP theorem, etc. We will also continuously keep in mind important additional topics that invariably arise in real world applications of Big Data, such as budgeting, compliance, etc.. Students are required to bring a laptop to class every week.

Terms Offered: Autumn

Prerequisite(s): MPSC 51036 or 51040 or 51042 or 51043 or 51100 or CAPP 30122 or MACS 30122. Very basic programming skills in Java. Basic linux IT skills.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPSC 53020. Foundations of Database Systems. 100 Units.

The course will cover the foundations of Database Management Systems (DBMS). This includes data models, database design, SQL, query processing, NewSQL/NoSQL, and systems for data analytics. The course will include 4 JAVA programming projects that implement and evaluate core database system components.

Terms Offered: Autumn

Prerequisite(s): In order to register for this course you must have taken MPSC 51036 or 51040 or 51042 or 51046 or 51100 or take concurrently.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPSC 53110. Foundations of Computational Data Analysis. 100 Units.

Foundations of Computational Data Analysis covers mathematical prerequisites for the Data Analytics Specialization courses in machine learning, and large-scale data analytics (MPSC 53111 and 53112): basic statistics and linear algebra. Topics in statistics include discrete and continuous random variables, discrete and continuous probability distributions, variance, covariance, correlation, sampling and distribution of the mean and standard deviation of a sample, central limit theorem, confidence intervals, maximum likelihood estimators, and hypothesis testing. Topics in linear algebra include Gaussian elimination, matrix transpose and matrix inverse, eigenvectors and eigenvalues, and singular value decompositions. In some of the exercises we'll use Python to compute and/or visualize data.

Terms Offered: Winter

Prerequisite(s): B+ or above in MPSC 51042 Python Programming, B+ or better in any other Core Programming class with prior knowledge of Python, or Core Programming waiver. B or above in MPSC 55001 Algorithms *If you are a student who is in MPSC 51013 this Autumn quarter and you would like to take Foundations along with Algorithms in the Winter quarter, we require that you earn at least a B+ in MPSC 51013. The course load for both of these classes is extremely high and taking them together will be very challenging. This course requires competency in Unix and Linux. If you attended the MPSC Unix Bootcamp you covered the required material. If you did not, please review the UChicago CS Student Resource Guide here: <https://uchicago-cs.github.io/student-resource-guide/>.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPSC 53111. Machine Learning. 100 Units.

This course introduces the fundamental concepts and techniques in data mining, machine learning, and statistical modeling, and the practical know- how to apply them to real-world data through Python-based software.

The course examines in detail topics in both supervised and unsupervised learning. These include linear and logistic regression and regularization; classification using decision trees, nearest neighbors, naive Bayes, boosting, random trees, and artificial neural networks; clustering using k-means, expectation-maximization, hierarchical approaches, and density-based techniques; and dimensionality reduction through PCA and SVD. Students use Python and Python libraries such as NumPy, SciPy, matplotlib, and pandas for implementing algorithms and analyzing data.

Terms Offered: Spring

Prerequisite(s): 1. B+ or above in MPSC 51042 Python Programming (or in Programming core requirement with prior knowledge of Python) 2. B+ or above in MPSC 55001 Algorithms 3. B or above in MPSC 53110

Foundations of Computational Data Analysis (or Data Analysis placement exam) If you are concurrently taking Algorithms with Machine Learning, a B+ or higher in MPSC 51013 Math for Computer Science If your grades in the above classes do not meet the minimum requirements set above, please contact the instructor to discuss your background.

Note(s): This course is not open to non-MPCS students.

MPSC 53112. Advanced Data Analytics. 100 Units.

In this course we study the algorithms and the associated distributed computing systems used in analyzing massive datasets, or big data, and in large-scale machine learning. We also cover the foundations of reinforcement learning. We focus on two fundamental ideas for scaling analysis to large datasets: (i) distributed computing, and (ii) randomization. In the former, we study how to design, implement, and evaluate data analysis algorithms for the distributed computing platforms MapReduce/Hadoop and Spark. In the latter, we explore techniques such as locality sensitive hashing, Bloom filters, and data stream mining. They are the foundation of modern data analysis in companies such as Google, Facebook, and Netflix. Reinforcement learning refers to the situation in which you want to model your environment, but you don't have a data set for training. Instead you learn by interacting with your environment. We'll learn algorithms that, e.g., teach themselves how to play chess by simply playing the game (against another copy of themselves) millions of times! They have applications in autonomous systems, robotics, operations research, responsive website design, stock trading, etc.

Prerequisite(s): MPSC 50101 Math for Computer Science MPSC 55001 Algorithms MPSC 51042 Python Programming (or Programming core requirement with prior knowledge of Python) MPSC 53110 Foundations of Computational Data Analysis MPSC 53111 Machine Learning In all the above courses a grade of B+ or above is

required. Please contact the instructor if you have, instead, equivalent courses or experience, or meet most but not all of the requirements.

Note(s): This class is not open to non-MPCS students.

MPCS 53113. Natural Language Processing. 100 Units.

Natural language processing (NLP) is the application of computational techniques, particularly from machine learning, to analyze and synthesize human language. The recent explosion in the amount of available text data has made natural language processing invaluable for businesses, social sciences, and even natural sciences. In this course we study the fundamentals of modern natural language processing, emphasizing models based on deep learning. These include language models, word embeddings, recurrent neural networks (Simple RNNs, LSTMs), context-free grammars and syntactic parsing, convolutional neural networks (CNNs), and attention-based models such as the Transformer, BERT, and the GPT models. We use Python and the deep learning framework PyTorch to build models from scratch and apply them to tasks such as machine translation. We also use the library of pre-trained models HuggingFace to build even more sophisticated models that can, for example, automatically generate summaries of articles. A significant component is the course project in which students build and train models to solve a real-world problem of their choice.

Terms Offered: Summer

Prerequisite(s): MPCS 50103 Math for Computer Science, MPCS Programming core requirement, MPCS 53110 Foundations of Computational Data Analysis, MPCS 53111 Machine Learning Equivalent courses or experience will be accepted with instructor permission.

Note(s): This class is not open to non-MPCS students.

MPCS 53120. Applied Data Analysis. 100 Units.

This course provides a self-contained introduction to computational data analysis from an applied perspective. It is intended as a standalone course for students who are not pursuing the full data analysis sequence in the MPCS. As such, students who have taken MPCS 53110 Foundations of Computational Data Analysis and received a grade of B or higher should take MPCS 53111 Machine Learning. Students that have taken or are currently enrolled in MPCS 53111 Machine Learning cannot register for this class. The course will cover topics in basic probability theory, statistical inference, and basic machine learning models typically used in data analysis. Each topic will be accompanied by example illustrations using Python. Many of the topics covered form the basis of almost all algorithms and machine learning methods used in data analysis. As an applied course, the emphasis will be on the use of these tools to solve problems.

Prerequisite(s): MPCS 50103 and MPCS 51036 or 51040 or 51042 or 51046 or 51100

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>

MPCS 53810. Topics in Algorithmic Game Theory. 100 Units.

'Intersection of game theory and computer science': analyzing/designing algorithms in strategic environments (where agents are strategic about what inputs they give the algorithm; they may lie for their own benefit!) 1.) Analysis: given the currently implemented algorithms (mechanisms), analyze participating agents' incentives/behavior. 2.) Design: design algorithms (mechanisms) that align participants incentives with 'overall welfare' and also have good algorithmic properties. Algorithmic mechanism design Learning goals: 1.) Be able to use fundamental game theoretic concepts to analyze how a 'system'/'mechanism' can incentivize participants to behave a certain way. 2.) Learn how to think about/ manage what the 'system' (mechanism) you are designing incentivizes participants to do. Aligning individual incentives with the 'overall good'. Examples: Classical/ more modern mechanisms. 3.) Gain exposure to the wide interplay between game theory and other computer-science topics: online algorithms, learning, optimization, cryptography, networks, routing, etc. 4.) Build widely useful skills: critical thinking and analysis/ independently reviewing literature, doing a project, and presenting your work and ideas to others.

Terms Offered: TBD

Prerequisite(s): MPCS 50103 Discrete Math (Immersion Math) OR a passing score on the Mathematics Placement exam. Core Programming (completed or concurrently taking).

MPCS 55001. Algorithms. 100 Units.

The main focus of this course is on the design and analysis of efficient algorithms, through the study of various algorithmic problems in different topics. Weekly homework assignments in theory and programming develop strong problem-solving skills. In addition to solving interesting homework problems employing various algorithmic techniques, students learn to implement their algorithmic ideas in python to solve a programming problem each week. At the end of the course the theory of NP-completeness is introduced, and some of the many interesting and important problems for which we do not have efficient algorithms are presented. The course goal is to teach students the skills to design new algorithms for simple problems and to understand both mathematical and applied aspects of algorithm efficiency and correctness.

Terms Offered: Autumn Spring Winter

Prerequisite(s): MPCS 50103 Math for Computer Science OR successfully passing the MPCS Mathematics Placement exam. MPCS 51036 or 51040 or 51042 or 51046 or 51100 (completed or concurrently taking).

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 55005. Advanced Algorithms. 100 Units.

Advanced Algorithms is a course on the design and analysis of efficient algorithms on a series of challenging topics. This course presents many interesting and relevant algorithms and data structures and gives students the tools to recognize and rigorously solve algorithmic problems often encountered in industry and engineering. Topics include: -Intractability. Theory of NP Completeness. Problems and Reductions: Hamilton Cycle to Traveling Salesman, Vertex Cover to Independent Set to Clique, SAT to 3SAT, 3SAT to Vertex Cover, Circuit-SAT to SAT. -Approximation algorithms. Metric k-center problem, Metric traveling salesman problem. Set Cover problem. -Online and streaming algorithms. Online counting algorithms. Potential functions. Heavy-hitters algorithms. Sketching algorithms. Multiplicative weights algorithm. -Advanced data structures. Skip lists, Segment trees, Tries, Advanced hashing. -String algorithms. KMP algorithm. Rabin's rolling hash fingerprinting algorithm. Aho-Corasick algorithm. -Advanced linear programming. LP duality. Integer linear programming and rounding. -Advanced dynamic programming. Knuth optimization and divide-and-conquer optimization. -Computational geometry. Convex hull algorithms. Line-segment intersection algorithm. Voronoi Diagrams and Delaunay Triangulations. -Optimization.

Terms Offered: TBD

Prerequisite(s): B+ or better in MPCS 55001 Algorithms or MPCS 55003 Intermediate Algorithms *or consent of instructor* This course requires competency in Unix, Linux, and GitHub. If you attended the MPCS Unix Bootcamp you covered the required material. If you did not, please review the UChicago CS Student Resource Guide here: <https://uchicago-cs.github.io/student-resource-guide/>.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 56430. Introduction to Scientific Computing. 100 Units.

This course aims to introduce students to important concepts in scientific computing. Students will survey topics in variety of disciplines and explore how rigorous computation is transforming the way that research conducted, experiments are executed, and data is analyzed across all scientific disciplines. The course will focus on applying a variety of strategies to solve research problems with a focus on scalable, reproducible, and collaborative work. This will include the creation, manipulation, and analysis of data sets and the application of complex algorithms. Students will design and implement systems that are reliable, capable of handling vast amounts of data, and utilize best practices in interface and usability design. Students will also learn to how to effectively communicate the results of their work to different audiences through written materials and presentation and to develop and provide resources for other researchers to use. While this course should be of interest for students interested in scientific research, techniques and approaches taught will be applicable to other fields.

Terms Offered: Autumn

Prerequisite(s): MPCS 50103 and MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 56511. Introduction to Computer Security. 100 Units.

This course introduces security principles and practices. Topics will range from encryption to network, endpoint, and application threats and controls. The course will emphasize both offense (i.e., attacker mindset) and defense (the importance of implementing both technical and non-technical controls). Topics: • Threat Landscape • Encryption • Authentication and access control • Endpoint security • Network security • Web and application security • Risk management • Monitoring and vulnerability management • Incident response Familiarity with Linux command line (required) Familiarity with TCP/IP and network routing (recommended) Students must be able to run the labs -- either on their own machine (virtual machine) or via a cloud-based option that requires installing an SDK on the local machine. Installation will be reviewed on week 1 - nothing to do in advance. This does not cost. This course requires basic competency in Unix and Linux. If you attended the MPCS Unix Bootcamp you covered the required material. If you did not, please review the UChicago CS Student Resource Guide here: <https://uchicago-cs.github.io/student-resource-guide/>.

Terms Offered: Autumn Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122 or MACS 30122.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 56520. Advanced Security Topics. 100 Units.

Knowing how to defend means knowing how to attack. This course will focus on penetration testing, drawing on the core concepts from the 'Introduction to Computer Security' course. Topics will include: passive reconnaissance and OSINT, active reconnaissance and enumeration, scanning techniques, vulnerability assessment, exploitation techniques, privilege escalation, and lateral movement. Labs will primarily use Kali linux.

Terms Offered: Spring

Prerequisite(s): MPCS 51036 or 51040, or 51042, or 51046, or 51100, or CAPP 30122 and MPCS 56511. Students need to be able to run both Kali linux and the Metasploitable2 machine image - on VirtualBox or other virtualization environment. MPCS 54001 Networks (recommended) or similar solid knowledge of TCP/IP, routing, and other networking concepts. MPCS 53001 Databases (recommended) or similar understanding of relational database structure and SQL (structured query language). This course requires competency in Unix

and Linux. Please plan to attend the MPCs Unix Bootcamp (<https://masters.cs.uchicago.edu/page/mpcs-unix-bootcamp>) or take the online MPCs Unix Bootcamp Course on Canvas.

Note(s): Non-MPCs students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCs 56535. Foundations of Cryptography. 100 Units.

This is a proofs-based theory course where we will study the foundations of various cryptographic primitives and encryption schemes. Topics could include (some of): -Formal notions of secrecy: information theoretic, perfect secrecy, computational secrecy, chosen plaintext attack (CPA), chosen ciphertext attack (CCA) etc. - One Way Functions, Pseudo Random Generators -Symmetric Key Cryptosystems -Public Key (Asymmetric) Cryptosystems: Examples. Primality testing based: RSA, Discrete-log based: Diffie-Hellman protocol -Message Authentication Codes -Digital Signatures (for example RSA based) -Secure Multi-Party Computation -Hash Functions, collision resistance, birthday attacks -Hard Core Predicates, Goldreich-Levin Theorem -Differential Privacy -Zero knowledge proofs: How to prove you solved a problem without giving away the solution; proof of knowledge/ proof of work, with some interesting applications to blockchains -Lattice Based Cryptography: post-quantum cryptography

Prerequisite(s): A- or above in MPCs 50103 Math for Computer Science: Discrete Mathematics OR successfully passing the MPCs Mathematics Placement exam. Core Programming (completed or concurrently taking).

Note(s): Non-MPCs students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCs 56540. Software Quality Assurance. 100 Units.

This course introduces techniques and standards for achieving and ensuring high quality in computer software. It includes a discussion of software quality and metrics that can be used to assess it; the activities that are required to establish an effective quality assurance approach including a variety of relevant standards, and the process related issues that must be implemented to achieve continuous quality improvements.

Terms Offered: Summer

Prerequisite(s): MPCs 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122. Students should know at least one common programming language, such as JavaScript, Python, Java, or C++. Familiarity with software development tools such as version control is expected. This course requires competency in Unix, Linux, and GitHub. If you attended the MPCs Unix Bootcamp you covered the required material. If you did not, please review the UChicago CS Student Resource Guide here: <https://uchicago-cs.github.io/student-resource-guide/>.

Note(s): Non-MPCs students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCs 56550. Introduction to DevOps. 100 Units.

This course provides an in-depth introduction to DevOps principles, practices, and tools for modern software development and operations. Students will learn how to integrate development (Dev) and operations (Ops) to enhance software delivery, automation, and scalability. The course covers continuous integration (CI), continuous deployment (CD), infrastructure as code (IaC), monitoring, cloud-native applications, and security. By the end of this course, students will have hands-on experience with tools like Docker, Kubernetes, Terraform, Jenkins, GitHub Actions, Prometheus, and AWS/Azure/GCP.

Prerequisite(s): MPCs 51036 or 51040 or 51042 or 51046 or 51100 OR instructor consent. Basic knowledge of Linux/Unix systems Familiarity with Git version control Knowledge of scripting languages (Python, Bash, etc.)

Note(s): Non-MPCs students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCs 56600. Introduction to Blockchain. 100 Units.

This course is a comprehensive technical introduction to relevant topics in the wider ecosystem surrounding blockchain. Our technological focus will include substantive topics in fundamental problems that blockchain is attempting to solve (and is generating), including algorithms, cryptography, security and trust, autopoietic peer-to-peer networking, distributed ledgers, double spending, proof of work and ownership issues, decentralized applications, smart contracts, and supporting technologies. With that said, this is not a course in economics or monetary theory, trading cryptocurrencies, nor is it a course on regulatory or legal issues surrounding blockchain, although we will touch on many of these topics throughout the course. We will also cover broader applications of blockchain technology beyond cryptocurrencies and ICOs including use cases from finance, insurance, science, healthcare, pharmaceuticals. We will cover cryptocurrencies and bitcoin and mining as well as ethereum initially as our introduction to the problem space, but will quickly move on to building our own blockchain application. Students may leverage a number of technologies including containerization (Docker), as well as the MEAN stack or Ruby on Rails for the more ambitious. Students may work in whatever languages they know best and make the most sense in context. These may include Java/javascript, C++, python, ruby, C#, and others.

Prerequisite(s): MPCs 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122

Note(s): Non-MPCs students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCs 56640. Software Testing. 100 Units.

As programmers, we write a lot of code. How do we know that it works? One approach, taken by advocates of strong type systems, is to write programs that study the code very closely, analyze the data and operations

inside it and prove the program is free of most errors. The other approach is to simply run the code and see if it performs as expected. This class is about the second approach to verifying software. We'll write and run all sorts of software tests, from simple unit tests to complex automatic explorations of a program's input space. We'll cover the theory and practice of testing complex, mission-critical software such as relational database implementations. Our goal is that students will become so fast and effective at writing automated tests for their code, that they'll start unit testing the homework in all their future classes. Topics covered: -Approaches to software quality: static versus dynamic approaches -Unit test, input space partitioning, boundary value analysis -Test-driven development -Environment management and "functionalizing" programs -Test case generation with generative AI -In-code assertions -Property-based testing: property definitions, oracles and shrinking -Fuzz testing and testing in computer security -Testing highly complex software: invariants and formal models

Terms Offered: TBD

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122. Familiarity with Python and simple UNIX command line tools is required. This course requires competency in Unix, Linux, and GitHub. If you attended the MPCS Unix Bootcamp you covered the required material. If you did not, please review the UChicago CS Student Resource Guide here: <https://uchicago-cs.github.io/student-resource-guide/>.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 57002. Independent Study. 100.00 Units.

TBD

Terms Offered: Autumn Spring Summer Winter

MPCS 57005. MPCS - Independent Study. 100 Units.

Please speak with the MPCS administration for details.

Instructor(s): Andrew Binkowski Terms Offered: Autumn

Note(s): Approval from the MPCS administration is required for an independent study

MPCS 57010. MPCS-Practicum. 100 Units.

This course is meant for MPCS students only. As part of its course offering, the MPCS gives students the option of doing a practicum under the supervision of a faculty or staff member (known as the practicum advisor). This practicum can be counted as elective credit towards the student's Masters degree. During a practicum, a student must develop a well-defined project requiring roughly 100 hours of work throughout a single academic quarter (i.e., an average of 10 hours per week). Throughout the year, the MPCS seeks project proposals from faculty and staff members interested in working with Masters students. These proposals are distributed to our students, who must then apply to work on a specific project.

Instructor(s): Andrew Binkowski and Borja Sotomayor

Note(s): This class is not open to non-MPCS students

MPCS 57020. Application Development Capstone. 100 Units.

The MPCS Application Development Capstone allows students to work on developing a mobile or web app over the course of a quarter, under the supervision of a faculty advisor. Participation in the capstone is by application only; students must submit a project proposal that, ideally, continues the work they started in an application development class. During the capstone, students are required to submit a number of progress reports, and will present their app at a capstone showcase at the end of the quarter.

Terms Offered: TBD

Prerequisite(s): MPCS students only

Note(s): Approval from the MPCS Administration.

MPCS 57200. Generative AI. 100 Units.

Recent generative AI systems such as ChatGPT, Dall-E, and GitHub Copilot have shown its applicability to a wide range of problem domains, and it is quickly becoming a valuable tool in the software developer's toolbox. This course will provide an experienced software developer with no experience in Machine Learning with: - A conceptual understanding of the basic concepts of Generative AI -How to build Generative AI systems to generate output targeting their domain of interest -How to deploy and integrate Generative AI models into production applications

Terms Offered: TBD

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100 or CAPP 30122. Basic familiarity with Python. We do not assume any background in machine learning or statistics.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 57300. Data Science Clinic I. 100 Units.

In order to enroll in this class, students must first submit an application and be matched with a project. Visit the Data Science Clinic site for application deadlines, how to apply, and information session details: <https://github.com/dsi-clinic/the-clinic>. The Data Science Clinic partners with public interest organizations to leverage data science research and technology to address pressing social and environmental challenges. The Clinic also provides students with exposure to real-world projects and problems that transcend the conventional classroom experience including: working with imperfect datasets, applying models and algorithms to real-world data, navigating security and privacy issues, communicating results to a diverse set of stakeholders (e.g., industry, public interest, government agencies), and translating information into actionable insights, policy briefs and

software prototypes. The Clinic is an experiential project-based course where students work in teams as data scientists with real-world clients under the supervision of instructors. Students will be tasked with producing key deliverables, such as data analysis, open source software, as well as final client presentations, and reports.

Instructor(s): N. Ross Terms Offered: Autumn Spring Winter

Prerequisite(s): DATA 12000, DATA 13600, DATA 21100, DATA 21200, DATA 23100 (or equivalent) and by permission of instructor

Equivalent Course(s): CAPP 30300, MACS 30300, DATA 27100, PPHA 30581

MPCS 58020. Time Series Analysis and Stochastic Processes. 100 Units.

Stochastic processes are driven by random events. They can be used to model phenomena in a broad range of disciplines, including science/engineering (e.g. computational physics, chemistry, and biology), business/finance (e.g. investment models and operations research), and computer systems (e.g. client/server workloads and resilience modeling). In many cases relatively simple stochastic simulations can provide estimates for problems that are difficult or impossible to model with closed-form equations. In this class we focus on the rudimentary ideas and techniques that underlie stochastic time series analysis, discrete events modeling, and Monte Carlo simulations. Course lectures will focus on the basic principles of probability theory, their efficient implementation on modern computers, and examples of their application to real world problems. Upon completion of the course, students should have an adequate background to quickly learn in depth specific Monte Carlo approaches in their chosen field of interest.

Terms Offered: Spring Summer

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51043 or 51046 or 51100 and MPCS 50103 or receive a passing score on the MPCS math placement exam. Languages: Required: familiarity with C/C++, Python, or Java (other language options are acceptable, but consult instructor first). Recommended: C and Python.

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online <https://mpcs-internal.cs.uchicago.edu>.

MPCS 65000. Reading and Research. 100 Units.

TBD

