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

- A graduate professional curriculum leading to the Master of Science (S.M.) degree, for students who wish to enter or advance themselves in computer science practice. This is the MPCS program outlined below.
- A graduate research curriculum leading to the Ph.D. degree that prepares students to perform advanced basic research in computer science either in industry or academia. For more information on the Ph.D. program, please see the listing Department of Computer Science (http://collegecatalog.uchicago.edu/graduate/departmentofcomputerscience/).

The Masters Program in Computer Science (MPCS) (https://masters.cs.uchicago.edu) offers a comprehensive and professionally-oriented computer science education that combines the foundations of computer science with the applied and in-demand skills necessary for careers in technology. The MPCS is especially well suited for students interested in Data Analytics, 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 courses in Software Engineering, Big Data, Data Analytics, Machine Learning, High Performance Computing, Application Development, Web Development, Cloud Computing and Information Security.

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

**MS in Computer Science Full-Time**
The full-time Masters Program in Computer Science offers a professionally-oriented computer science education that combines the foundations of computer science with applied technical coursework. The full-time MPCS is especially well suited for students interested in Data Analytics, High Performance Computing, Application Development and Software Engineering. Full-time students at the University of Chicago take 3 classes per quarter and have the choice to complete the 9-Course program in one academic year or the 12-Course specialization program in 15 months. Daytime classes are available for full-time students.

**MS in Computer Science Part-Time** (https://masters.cs.uchicago.edu/page/ms-computer-science-part-time/)
The part-time Masters Program in Computer Science offers working professionals the opportunity to pursue a professionally-oriented computer science education that combines the foundations of computer science and applied technical coursework. This program allows the flexibility to complete the program, and enhance your technology skillset, at your own pace. Part-time students at the University of Chicago can take 1 to 2 classes per quarter, with most students completing the 9-Course Program in 18 months to 2 years. Evening classes are available for part-time students.

**Joint MBA/MPCS Program** (https://masters.cs.uchicago.edu/page/mba-mpcs/) with the Booth School of Business
The Joint MBA/MPCS program meets today’s leading tech companies’ cross-functional demands of new employees. 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.

**Pre-Doctoral MS in Computer Science**
This program is a 12-course research-oriented masters program for students who want to explore computer science research. The Pre-Doc program is for full-time students with a CS background starting in the Autumn quarter.

**Introduction to Programming and Math for Computer Science (Discrete Math)**
Introductory coursework in programming and math (called immersion classes) are available to any admitted MPCS student. Students can complete one or both of these classes before beginning coursework in the MS in Computer Science Program.

Please see our website for admissions requirements and deadlines (https://masters.cs.uchicago.edu/page/admissions/). To view a complete list of course offerings, please visit the MPCS Course Catalog (https://masters.cs.uchicago.edu/page/mpcs-course-catalog-0/).

For inquiries or questions please email questions@cs.uchicago.edu.
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 roughly two or three programs of significance each week to learn foundational programming principles and practices for writing clean, readable code, and learning how think and solve problems like a computer scientist. Along with basic principles like procedural abstraction, recursion, and handling input and output, an emphasis will be placed on theories and principles of Object Oriented software design, analyzing algorithms and choosing appropriate data structures to solve problems.

Instructor(s): T. Andrew Binkowski Terms Offered: Autumn Winter

Note(s): Open only to MPCS students

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

This course is an introduction to discrete mathematics oriented toward computer science. The course emphasizes mathematical proof and problem solving, employed on a variety of useful topics: logic; proof by induction; counting, factorials, and binomial coefficients; discrete probability; random variables, expected value, and variance; recurrences; graphs and trees; basic number theory; asymptotic notation, and rates of growth. On completion of the course, students will have been trained to think about and absorb mathematical concepts, to solve problems requiring more than standard recipes, and express 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 data mining.

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.

This course will instruct students on the fundamentals of mobile application development using Apple’s iOS SDK. An introduction to the Objective-C programming language, including memory management, 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. User interface and application design considerations specific to mobile technologies will 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: Autumn Winter

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

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

MPCS 51031. Android Application Development. 100 Units.

After a quick introduction to mobile computing, competing platforms, Android architecture, market projections, and social and economic implications, we will dive directly into developing several reference implementations. Alternating between theory and practice, and progressing cumulatively, will cover major feature of the Android platform, including: audio, graphics, internet connectivity, wifi, mapping/geo-positioning, notifications, sms, structured feeds, persistence, threads, states, and inter-process communication, among others. Students will choose a final project, then envision, design, develop, test, and deploy an application to the Android marketplace. Along with basic principles like procedural abstraction, recursion, and handling input and output, an emphasis will be placed on theories and principles of Object Oriented software design, analyzing algorithms and choosing appropriate data structures to solve problems.

Instructor(s): Adam Gerber Terms Offered: Spring

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51043 or 51100 and experience programming in Java

Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online https://masters.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 and iPad. 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 and tools, such as Xcode, Interface Builder and Instruments. 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.

Instructor(s): T. Andrew Binkowski 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://masters.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://masters.cs.uchicago.edu

MPCS 51036. Java Programming. 100 Units.
This 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 basic fundamentals of OO design—inheritance, polymorphism, composition, etc. 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, OO design patterns, as well as certain Java libraries such as Swing. For their final-projects, students will develop a multi-threaded, arcade-style game. The course format is both lecture and lab. We will use use being given 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.
Instructor(s): Dries Kimpe Terms Offered: Autumn Spring
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://masters.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.
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://masters.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.
Instructor(s): Dries Kimpe Terms Offered: Autumn Spring
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://masters.cs.uchicago.edu

MPCS 51042. Python Programming. 100 Units.
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.
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://masters.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++11 language, which Bjarne Stroustrup, the inventor of C++, says “feels like a new language.” It also discusses how to workaround in old 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.

Instructor(s): Michael Spertus
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://masters.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 C++ experts use to write libraries that simultaneously provide the optimum in ease-of-use, abstraction, and performance. If you use C++ in your daily life, you and your team will see substantial benefits from understanding and using C++ at a deeper level.

Instructor(s): Michael Spertus
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://masters.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/Al, 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 module

Instructor(s): Paul Romano
Terms Offered: Autumn
Prerequisite(s): MPCS 50101 Concepts of Programming (or programming waiver) and approval from MPCS administration
Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online https://masters.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.

Instructor(s): Mark Shacklette
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://masters.cs.uchicago.edu

MPCS 51052. Advanced Python Programming. 100 Units.
Python is a general-purpose programming language that is used in many application areas, including data science, machine learning/Al, 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.

Instructor(s): TBA
Terms Offered: Spring
Prerequisite(s): MPCS 51042
Note(s): All non-MPCS students must meet the course prerequisites and complete the MPCS course request form for approval to register. (https://masters.cs.uchicago.edu/page/course-requests)

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 application development in two contexts: highly scalable (or so-called “web-scale”) web applications, and enterprise applications in a hybrid environment comprising both on-premises and cloud infrastructure. We will focus primarily on infrastructure and platform services, and will 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.

Instructor(s): Vas Vasiliadis  
Terms Offered: Spring Summer

MPCS 51087. High Performance Computing. 100 Units.
Parallel computing allows multiple processing units to work together simultaneously on a common task. For certain types of applications, parallelization can increase execution time in proportion to the number of computers or processors used. This is a huge advantage for applications which have performance and/or memory bottlenecks, such as one typically encounters in financial modelling, physics, engineering, or other applied science domains. This is a fast-paced applied programming course aimed at students with significant development experience in either C, C++, or FORTRAN (Java, Matlab, or Python are also possible, but not ideal). No prior knowledge of parallel computing is assumed. Students should, however, have both an interest and some previous experience in either algorithmic development, numerical methods, applied mathematics, or perhaps any physics or engineering-type discipline. A brief overview of parallel computing will be presented at the outset, but the course will be less on overview of HPC architectures and much more on focus on algorithmic implementation and performance tuning. The goal of the course it to give students experience in developing efficient, scalable (distributed memory) parallel algorithms appropriate for any system running an implementation of the Message Passing Interface (MPI). Assignments will be designed with some flexibility to allow students to explore applying parallel techniques to applications in their own field of interest.

Instructor(s): Andrew Siegel  
Terms Offered: Winter

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.

Instructor(s): Andrew Siegel  
Terms Offered: Autumn

MPCS 51132. Full Stack Software Engineering. 100 Units.
Full stack software engineering will focus on the fundamentals of developing applications using the Kotlin programming language, Android Studio, AWS, and Spring. Students will create their own AWS accounts and deploy both AWS-lambdas and Spring application servers to their own AWS instances. Throughout the course, students will develop projects of progressive complexity using the tools and technologies introduced in the course. In addition to mastering the fundamentals of Android development, students will learn protocol buffers, and best practices in devOps and full-stack software engineering.

Instructor(s): Adam Gerber  
Terms Offered: Spring Summer

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.

Instructor(s): Peter Vassilatos  
Terms Offered: Autumn Winter
MPCS 51030. Entrepreneurship in Technology. 100 Units.
The core theme for the Entrepreneurship in Technology course is that computer science students need exposure to the broad challenges of capturing opportunities and creating companies. Most of the skills required for this process have nothing to do with one’s technical capacity. We’ll explore creating a story, pitching the idea, raising money, hiring, marketing, selling, and more. Real-world examples, case-studies, and lessons-learned will be blended with fundamental concepts and principles. The course will involve a business plan, case-studies, and lessons-learned will be provided in order to facilitate the delivery.
Instructor(s): Vasilios Vasiliadis Terms Offered: Winter
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://masters.cs.uchicago.edu

MPCS 51035. User Interface and User Experience Design. 100 Units.
This course is eleven weeks in length and covers the elemental practices of user interface (UI) design, user experience (UX) design, and user research. The intention of the course is to provide an overview of the experience design field so that the student is empowered to practice design as well as effectively manage design. It is, however, more likely that the student will collaborate with others on the design of products and services.
Prerequisite(s): MPCS 51030 or MPCS 51031.
Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online https://masters.cs.uchicago.edu

MPCS 51040. Product Management. 100 Units.
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.
Instructor(s): Vasilios Vasiliadis
Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100.
his 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.
Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online https://masters.cs.uchicago.edu

MPCS 51050. 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 tools and techniques will be provided in order to facilitate the delivery.
Instructor(s): Peter Vassilatos Terms Offered: Winter
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://masters.cs.uchicago.edu

MPCS 51220. Applied Software Engineering. 100 Units.
In this course, we will explore practical techniques to solving modern software challenges. Topics include: Software quality control, Test-driven development, Domain-driven design, Measuring software quality, Architectural design patterns, Edge-free programming, Event streams, logging, and audit trails, Source control techniques for small teams with Git, Security and cryptography essentials, Continuous integration & deployment.
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://masters.cs.uchicago.edu

MPCS 51230. User Interface and User Experience Design. 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 tools and techniques will be provided in order to facilitate the delivery.
Instructor(s): Peter Vassilatos Terms Offered: Winter
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://masters.cs.uchicago.edu

MPCS 51240. Product Management. 100 Units.
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.
Instructor(s): Vasilios Vasiliadis
Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100.
his 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.
Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online https://masters.cs.uchicago.edu

MPCS 51250. Entrepreneurship in Technology. 100 Units.
The core theme for the Entrepreneurship in Technology course is that computer science students need exposure to the broad challenges of capturing opportunities and creating companies. Most of the skills required for this process have nothing to do with one’s technical capacity. We’ll explore creating a story, pitching the idea, raising money, hiring, marketing, selling, and more. Real-world examples, case-studies, and lessons-learned will be blended with fundamental concepts and principles. The course will involve a business plan, case-studies, and supplemental reading to provide students with significant insights into the resolve required to take an idea to market. Class discussion will also be a key part of the student experience.
Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100.
Note(s): If an undergraduate takes this course as CMSC 29512, it may not be used for CS major or minor credit. Non-MPCS students must receive approval from program prior to registering. Request form available online https://masters.cs.uchicago.edu
Equivalent Course(s): CMSC 29512

MPCS 51300. Compilers. 100 Units.
This class teaches the theory and practice of how to write a compiler, including lexical analysis, grammars, lexers and parsers, type checking, and code generation. For decades, compilers have been the most dynamic and challenging branch in computer science. The main part of this class will focus on providing the basics of the
different phases of compilation. Through the course, students will develop appreciation for the implementation strategies behind making an efficient and robust compiler.

Terms Offered: TBD

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://masters.cs.uchicago.edu

MPCS 51400. Functional Programming. 100 Units.

This course presents the functional programming paradigm, based on the idea of functions as first-class values that can be computed and operated on like other data. Functional languages provide great power of expression while maintaining simplicity, making it easier to write correct and maintainable software. Functional programming has gradually found new applications in areas like finance, telecommunications, and graphics. The essential feature of treating functions as values has also been added to a broad range of conventional languages, such as Python, C++, Java, Apple's Swift and Google's Go language. The course will use the Haskell language based on its representing a purely functional language and its large community support that helps with writing Haskell programs easily. After learning the basic elements of these languages, we will explore functional programming techniques that can be exploited in many areas of application. In particular, we will examine how FP features are used in more modern languages and libraries such as C++, Java, Elm, and React libraries and how they are used in real-world settings. We will briefly compare the functional paradigm with the related paradigm of object oriented programming. If time permits then we will explore more advanced topics including concurrent functional programming and functional reactive programming.

Instructor(s): Lamont Samuels

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51043 or 51046 or 51100. 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://masters.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/DDD). 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.

Instructor(s): J. Mark Shacklette

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://masters.cs.uchicago.edu

MPCS 52010. Computer Architecture. 100 Units.

This course focuses on the design and performance evaluation of modern computer architectures. The emphasis is on microprocessors, chip-multiprocessors and memory hierarchy design, particularly in the context of parallel (multicore) CPUs.

Instructor(s): Andrew Siegel

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

Terms Offered: Autumn Spring

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.

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

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

Instructor(s): Andrew Siegel

Terms Offered: Summer

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

Note(s): Non-MPCS students must meet the prerequisites and request approval - https://masters.cs.uchicago.edu/page/course-requests

MPCS 52030. Operating Systems. 100 Units.

This is an introductory course on operating systems. Students will learn the fundamentals of how modern operating systems are built, from the interface with hardware up through the kernel-userspace boundary. Important topics include the relationship between processes and threads, synchronization, inter-process communication, memory management, file systems, scheduling, I/O, virtualization. These concepts will be reinforced through several large-scale programming projects (in C++), whereby students will implement various sub-components of a real operating system. Prior experience with C and/or C++ required. As appropriate, we’ll use the Linux operating system (written in C) as an example of operating systems design. As time permits, we will also delve into current hot topics in the field (such as multi-core systems, security, and cluster/grid computing).

Terms Offered: Spring

Prerequisite(s): Student must meet one of the following requirements: - B+ or higher in MPCS 51040 - C Programming - B+ or higher in MPCS 51100 - Advanced Programming - A- or higher in MPCS 51044 - C++ for Advanced Programming (students who have only taken this class should review C before taking OS) Students must also have a B+ or higher in MPCS 52011 - Introduction to Computer Systems.

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

MPCS 52040. Distributed Systems. 100 Units.

This class teaches the theory and practice of how to design a distributed system. Cloud computing, drive services, online collaborative working environment, massively multiplayer online gaming, airline reservation systems, e-commerce are examples of distributed systems. Because of a continuous growing number of independent and diverse connected devices, the need of managing heterogeneous architecture became prominent. We will learn how to create a reliable system being easy to use but handling the complexity of having all these machines work collectively. The objective of this class is to learn distributed system by studying: Communication mechanisms Synchronization Scalability Consistency / Replication Fault tolerance

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://masters.cs.uchicago.edu

MPCS 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): MPCS 51036 or 51040 or 51042 or 51046 or 51100. Familiarity with C, Java, and/or Python.
MPCS 52553. Web Development. 100 Units.
This course provides students with an introduction to modern web development, with an emphasis on the pragmatic skills needed to build live, functioning web applications. Students will learn fundamental domain modeling skills, HTML and CSS frameworks, agile software techniques and best practices, Javascript and AJAX, and both server-side and client-side debugging techniques. We will use the Ruby language and the Rails framework to immerse students into the challenge of building a live, database-backed web application deployed at a public web address. Specifically, students will learn how to: Build a live website or web application and deploy it to the public internet; Use the Ruby on Rails framework to rapidly build a web application; Write software using the Ruby programming language; Use a relational database to provide content for dynamic websites; Follow industry best-practices of modern web software development; Troubleshoot and resolve the most common problems with web applications
Term(s) Offered: Spring Winter
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://masters.cs.uchicago.edu

MPCS 52554. Advanced Web Development. 100 Units.
This course builds upon MPCS 52553 to enable students to gain mastery over modern web architectures and services. Today’s consumer-facing and business applications must consume external services and publish services of their own. Students will build interconnected chains of services, with a particular emphasis on efficiency, security, and sustainability using modern web frameworks such as Rails, React, Node, and more.
Instructor(s): Jeffrey L Cohen Terms Offered: Autumn
Prerequisite(s): MPCS 52553 Web Development
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 utilizes software to tackle real-world problems in the world of finance. Through hands-on modules, we will focus on two main financial assets 1) public equity (stock) research and trading strategies and 2) fixed income research and valuation through the analysis of mortgage loans. The class will expose students to equal parts financial theory and technology using real world data. Technologies will include SQL database, Python dataframes, columnar datasets and javascript. Baseline code will be provided and students will be expected to improve upon through their own efforts and creativity to deliver two projects, including their own stock trading algorithm. Given the format, students should be comfortable building upon provided code and working to improve upon it, specifically for databases and scripting languages. Students should also be motivated by the open-ended nature of the projects that will require an interest in learning and applying financial theory in real life settings.
Term(s) Offered: Autumn
Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100
Note(s): Non-MPCS students must receive approval - https://masters.cs.uchicago.edu/page/course-requests

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.
Term(s) Offered: Autumn Spring
Prerequisite(s): MPCS 50101 or receive a passing score on the MPCS programming waiver.
Note(s): Non-MPCS students must receive approval from program prior to registering. Request form available online https://masters.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,
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 forests, 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 analyzing and visualizing datasets.

Instructor(s): Amitabh Chaudhary
Terms Offered: Winter

Prerequisite(s): This course requires mathematical, algorithmic, and programming maturity. Specific course prerequisites are: MPCS 50101 Math for Computer Science, MPCS 55001 Algorithms, MPCS Programming core requirement. In each of the above courses a B+ or better grade is required. Equivalent courses will be accepted with instructor permission. In addition, students are expected to be familiar with — Programming in Python: use of lists, dictionaries, conditional classes, and reading from and writing to files. Data structures: such as trees and graphs. Basic multivariate calculus: including differentiation, integration, and finding maxima and minima. Basic Linear Algebra: vectors, matrices, matrix multiplication, linear transformations, and eigenvectors. If you are unfamiliar with just one or two topics, you may be allowed to take the course if you are committed to learning those on your own. In that case, or if you have other questions, please email the instructor.

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

MPCS 53111. Machine Learning. 100 Units.

This course explores selected advanced themes in data mining and analytics. These include the recent "model-free" techniques for mining massive datasets, foundations of natural language processing, and time series analysis. Topics include frameworks such as MapReduce; algorithmic ideas such as locality-sensitive hashing, Bloom filters, random walks, and competitive analysis; and applications such as link analysis, social-network analysis, recommendation systems, streaming data, and advertising on the web. In natural language processing, the course introduces fundamentals of language models, text classification, and information retrieval and extraction. In time series analysis, the course examines stationary processes and the ARIMA and GARCH models.

Instructor(s): Amitabh Chaudhary
Prerequisite(s): MPCS 50101 Math for Computer Science MPCS 55001 Algorithms MPCS 51042 Python Programming (or Programming core requirement with prior knowledge of Python) MPCS 53110 Foundations of
Computational Data Analysis MPCS 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.
Can we predict how people will vote based on their twitter conversations? Can we identity pairs of researchers who will benefit from collaborating with each other based on their published articles? In this course we will study techniques for automatically detecting patterns and learning hidden structures in text data. Such techniques are of tremendous value due to the explosion in the amount of available text data, and their potential benefit to social sciences and businesses. We will learn the fundamental steps in the natural language processing, such as syntactic parsing or understanding the structure of a sentence, and semantic analysis or understanding the meaning of a sentence from the meanings of the words in it. These will help us build sophisticated models for text classification, such as for detecting sentiment or identifying fake news. We will see that the primary challenge is that natural languages are ambiguous. For instance, the sentence I made her duck can be interpreted in five different ways! So our models are probabilistic, and we resolve the ambiguity by training on large amounts of text corpora. We will study a variety of models in the context of text processing including Markov and hidden Markov models, naive Bayes, logistic regression, and neural networks. All through the course we will use Python and libraries such as the Natural Language Toolkit (NLTK) for processing real-world data.

Instructor(s): William Connor 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 do not want to pursue the full data analysis sequence in the MPCS. As such, students who have taken or are taking MPCS 53111 Machine Learning cannot register for this class. Students who have taken MPCS 53110 Foundations of Computational Data Analysis must obtain MPCS administration approval before registering 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 computational packages and software. Many of the topics covered form the basis of almost all algorithms and machine learning methods used in big data analysis. Emphasis will be given on using these techniques for problem solving. All work will be done in R.

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://masters.cs.uchicago.edu

MPCS 54001. Networks. 100 Units.
Broadly, this course will focus on the history, theory and implementation of computer networks. We will discuss the low-level technologies that move bits around (such as Ethernet and WiFi), the high-level applications that are part of our everyday 21st-century lives (such as email, the Web, and mobile phones), and everything in between (security, TCP/IP). At the completion of this quarter, you will (or should!) be able to explain, in detail, how data makes it way around the Internet when you click on a web link, how you can drive around at 80 MPH talking on a cell phone without the call dropping, how you can make a streaming video call over a lossy wireless link without frame dropping or jitter. In short, we’ll pull back the curtain on what can be a somewhat mysterious and magical part of working with computers.

Instructor(s): William Connor Terms Offered: Autumn Winter
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://masters.cs.uchicago.edu

MPCS 55001. Algorithms. 100 Units.
The course is an introduction to the design and analysis of efficient algorithms, with emphasis on developing techniques for the design and rigorous analysis of algorithms rather than on implementation. Algorithmic problems include sorting and searching, discrete optimization, and algorithmic graph theory. Design techniques include divide-and-conquer methods, dynamic programming, greedy methods, graph search, as well as the design of efficient data structures. Methods of algorithm analysis include asymptotic notation, evaluation of recurrences, and the concepts of polynomial-time algorithms. NP-completeness is introduced toward the end of the course. Students who complete the course will have demonstrated the ability to use divide-and-conquer methods, dynamic programming methods, and greedy methods, when an algorithmic design problem calls for such a method. They will have learned the design strategies employed by the major sorting algorithms and the major graph algorithms, and will have demonstrated the ability to use these design strategies or modify such algorithms to solve algorithm problems when appropriate. They will have derived and solved recurrences describing the performance of divide-and-conquer algorithms, have analyzed the time and space complexity of dynamic programming algorithms, and have analyzed the efficiency of the major graph algorithms, using asymptotic analysis.

Terms Offered: Autumn Spring
MPCS 5003. Intermediate Algorithms. 100 Units.

The course begins with an in-depth study of computational intractability and NP-completeness, and follows it by studying practical algorithms for intractable problems: approximation algorithms and those based on local search. It then looks at how the power of random choices can be harnessed to avoid worst-case situations. The resulting randomized algorithms have been crucial in the success of modern computer systems. The next topic is amortized analysis, an advanced technique used to analyze situations in which algorithms maybe expensive in some of their operations, but are provably efficient over a sequence of operations.

Terms Offered: TBD

Prerequisite(s): This course requires a strong command of discrete mathematics, including discrete probability, and introductory algorithms. For discrete mathematics, students should have taken MPCS 50103 Mathematics for Computer Science: Discrete Mathematics and obtained a B+ or higher, or passed the MPCS math placement exam. For introductory algorithms, students must take MPCS 55001 Algorithms and obtained a B+ or higher.

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

MPCS 55005. Advanced Algorithms. 100 Units.

This course introduces computer scientists to the field of bioinformatics. The vast amounts of data produced in genomics related research has significantly transformed the role of biological research. High-throughput automated biological experiments require advanced algorithms, implemented in high-performance computing systems, to interpret their results. We will focus on analyzing complex data sets in the context of biological problems. Students will design and implement systems that are reliable, capable of handling huge amounts of data, and utilize best practices in interface and usability design to accomplish common bioinformatics related problems. While this course should be of interest for students interested in biological sciences and biotechnology, techniques and approaches taught will be applicable to other fields. This course will present a practical, hands-on approach to the field of bioinformatics. The topics covered in this course will include: software, data mining, high-performance computing, mathematical models and other areas of computer science that play an important role in bioinformatics. Existing methods for analyzing genomes, sequences and protein structures will be explored, as well as computing infrastructure that support their efficient utilization. Students will be introduced to all of the biology necessary to understand the applications of bioinformatics algorithms and software taught in this course.

Instructor(s): Andrew Binkowski Terms Offered: Spring

Prerequisite(s): MPCS 53001 and 51036 or 51040 or 51042 or 51046 or 51100 (completed or concurrently taking).

Note(s): All non-MPCS students must meet the course prerequisites and complete the MPCS course request form for approval to register. (https://masters.cs.uchicago.edu/page/course-requests)

MPCS 56511. Introduction to Computer Security. 100 Units.

This course introduces computer security principles and practices. Topics will range from classical cryptography to recent web application security risks listed in the OWASP Top 10. The course will emphasize both offense (i.e., attacker mindset) and defense (i.e., designing and building secure systems). Topics: • Threat modeling • Cryptography • TLS and HTTPS • Web application security • Network security • Authentication and access
control ● Memory safety and isolation ● Trusted computing ● Side channels ● Anonymity and web privacy ● Underground economy ● Human factors ● Security ethics ● Cryptographic policy

Instructor(s): William Conner
Terms Offered: Winter

Prerequisite(s): MPCS 51036 or 51040 or 51042 or 51046 or 51100. MPCS 52011 recommended. Familiarity with C, Java, and/or Python (required). Familiarity with Linux command line (recommended, but not required) Non MPCS students must meet prerequisites and complete the request form http://tinyurl.com/mpcs-courseform.

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

MPCS 56520. Advanced Security Topics. 100 Units.
This course extends the 'Introduction to Computer Security' course to cover more advanced security topics, particularly from a security practitioner point of view. This includes a range of threats and defenses, organizational readiness concepts and best practices, and use of security tools.

Instructor(s): Shelley Rossell
Terms Offered: Autumn

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

Note(s): Non-MPCS students must meet the prerequisites and request approval - https://masters.cs.uchicago.edu/page/course-requests

MPCS 57002. Independent Study. 100 Units.
TBD
Terms Offered: Autumn Spring Summer Winter

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 58001. Numerical Methods. 100 Units.
This is a practical programming course focused on the basic theory and efficient implementation of a broad sampling of common numerical methods. Each topic will be introduced conceptually followed by detailed exercises focused on both prototyping (using matlab) and programming the key foundational algorithms efficiently on modern (serial and multicore) architectures. The ideal student in this course would have a strong interest in the use of computer modeling as predictive tool in a range of disciplines -- for example risk management, optimized engineering design, safety analysis, etc. The numerical methods studied in this course underlie the modeling and simulation of a huge range of physical and social phenomena, and are being put to increasing use to an increasing extent in industrial applications. After successfully completing this course, a student should have the necessary foundation to quickly gain expertise in any application-specific area of computer modeling.

Instructor(s): Andrew Siegel
Terms Offered: Spring

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.

Note(s): All non-MPCS students must meet the course prerequisites and complete the MPCS course request form for approval to register. (https://masters.cs.uchicago.edu/page/course-requests)

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): All non-MPCS students must meet the course prerequisites and complete the MPCS course request form for approval to register. (https://masters.cs.uchicago.edu/page/course-requests)
MPCS 65000. Reading and Research. 100 Units.
TBD