

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 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.
- 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.

The Masters Program in Computer Science (<http://csmasters.uchicago.edu>) (MPCS) 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 today's careers in technology. The MPCS is specially well suited for students interested in careers in software engineering, mobile computing, data analytics, and high-performance computing.

The coursework in our program 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, Mobile Application Development, Web Development and Cloud Computing.

What sets our program apart is our ability to tailor coursework to the career goals and backgrounds of our students. For students that do not have a background in math or programming, we offer math and programming prerequisite courses to introduce students to computing and to the fundamental and introductory skills that are needed to successfully begin masters-level coursework. Students with more advanced backgrounds can begin in higher-level classes.

The MPCS offers four Programs of Study (<https://csmasters.uchicago.edu/page/programs-study>) **and a number of specialization options:**

MS in Computer Science for those extending their computer science education and experience.

- **9-Course MS in Computer Science Program** - This program consists of 9 courses that students can complete in just 9 months (full-time) or in 15 months (part-time).
- **12-Course MS in Computer Science Specialization Program** - This program is designed for students seeking further specialization that will build a strong professional skill set in the following focused areas: Software Engineering, High Performance Computing, Data Analytics, Mobile Computing. This program can be completed in 15 months, if full time, including a summer internship.

MS in Computer Science (with immersion) for those starting a career in technology.

- This program is tailored for students who have no background in computer science, programming or discrete math. Taking Math and Programming prerequisite courses prepares students in this option to join the MS in Computer Science program and complete it successfully.

Pre-Doctoral MS in Computer Science for those wanting to pursue research.

- This 12-course research-oriented masters program is 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.

Joint MBA/MPCS Program for those looking to merge business and technology.

To view the specific requirements for each program and a complete list of course offerings, please visit the MPCS website at: <https://csmasters.uchicago.edu>

Masters Program in Computer Science Courses

MPCS 50101. Concepts of Programming (Prerequisite Programming Course) 100 - 150 Units.

In this course students will get an introduction to the field of computer science by learning to program in Java. 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): TBA Terms Offered: Summer; Winter

Note(s): Open only to MPCS students

MPCS 50103. Math for Computer Science: Discrete Math (Prerequisite Math Course) 100 - 150 Units.

This course in 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.

Instructor(s): Geraldine Brady Terms Offered: 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 51026. Trading Systems Design. 100 Units.

This class teaches the theory and practice of how to design a trading system. Since 1998, after the U.S. Securities and Exchange Commission authorized electronic exchanges, the financial world has been using computer science extensively. The High Frequency Trading became a main actor of the main exchanges across the world. During this course, we will learn how to create a reliable high-frequency trading system.

Instructor(s): Sebastien Donadio Terms Offered: Summer

Prerequisite(s): Core Programming

Note(s): Non-MPCS students need to complete a course request form.

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 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 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.

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

Prerequisite(s): MPCS 50101 or programming waiver

Note(s): Non-MPCS students must receive approval from program prior to registering.

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 every 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 chose a final project, then envision, design, develop, test, and deploy an application to the Android marketplace.

Instructor(s): Adam Gerber Terms Offered: Spring

Prerequisite(s): MPCS 51036 or equivalent experience programming in Java

Note(s): Non-MPCS students must receive approval from program prior to registering.

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.

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 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.

Instructor(s): Adam Gerber Terms Offered: Autumn, Spring

Prerequisite(s): MPCS 50101 or programming waiver

Note(s): Non-MPCS students must receive approval from program prior to registering.

MPCS 51037. Advanced Java Programming. 100 Units.

This is an advanced course designed for students with a good foundation in Java programming. Basic familiarity with C is also assumed. The course focuses on designing distributed, multithreaded applications with the Java platform. It is an application programming course. Emphasis is placed on applying technology rather than studying API design and implementation. Topics proceed (roughly) from "low-level" to high level network programming concepts; socket byte streams, object serialization, Remote Method Invocation, Java/CORBA (minimal), Web Services, and (briefly) Enterprise Java Beans. While any of these topics alone could form the basis for an entire course, the emphasis is on providing students with an adequate foundation for pursuing individual topics in greater depth. Along the same lines, a major focus of the course is to help students determine when to best apply a given Java technology in a real world, multi-tier application.

Instructor(s): Adam Gerber Terms Offered: Summer

Prerequisite(s): MPCS 51036 Java Programming or consent of instructor

Note(s): Non-MPCS students must receive approval from program prior to registering.

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): MPCS 50101 or programming waiver

Note(s): Non-MPCS students must receive approval from program prior to registering.

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 work around 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 Terms Offered: Winter

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

Note(s): Non-MPCS students must receive approval from program prior to registering.

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.

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): Core programming requirement including 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.

MPCS 51081. Unix Systems Programming. 100 Units.

MPCS 51081 is a UNIX systems programming course (as opposed to an operating systems course) that explores various topics in systems programming on the Unix platform. This course is NOT a course in operating systems development. We will focus on the Unix C APIs around file I/O, processes and signals, pipes, and System V interprocess communication. In addition to the traditional systems programming topics, this course will also introduce students to a significant level of detail in the use of Sun RPC (Remote Procedure Calls) and multithreaded programming, including operating system support and models (1-1, many-1, many-many). Students will be exposed to the numerous issues involved in safe and efficient multithreading strategies using the POSIX pThreads API as implemented by Linux's clone() and Solaris threads. Multithreading architectures will be discussed as well as advanced issues such as mutexes, semaphores, race conditions, deadlocks, etc. Berkeley socket programming will be covered in detail, as well as the creation and use of shared libraries. Various tools used in developing software in C on Unix will be covered, including gcc, gdb, ddd, gprof, cvs, etc. Because this is a programming course, students will be expected to know the C programming language upon course entry. There will be a C programming qualifying test taken during the first week of class for all students.

Instructor(s): Mark Shacklette Terms Offered: Winter

Prerequisite(s): MPCS 51040 C Programming and Unix Bootcamp

Note(s): Non-MPCS students must receive approval from program prior to registering.

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 Vasiladis Terms Offered: Spring, Summer

Prerequisite(s): MPCS 50101 or programming waiver

Note(s): Non-MPCS students must receive approval from program prior to registering.

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 a focus on algorithmic implementation and performance tuning. The goal of the course is 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

Prerequisite(s): MPCS 51040 C Programming

Note(s): Non-MPCS students must receive approval from program prior to registering.

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

Prerequisite(s): For students who have taken the programming immersion course, a minimum grade of A- plus 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.

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

Prerequisite(s): MPCS 50101 or programming waiver

Note(s): Non-MPCS students must receive approval from program prior to registering.

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.

Instructor(s): Benjamin Johnson Terms Offered: Spring

Prerequisite(s): Core Programming Requirement (MPCS 51036, 51040 or 51100)

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.

Instructor(s): Sebastien Donadio Terms Offered: TBD

Prerequisite(s): Core Programming Course (MPCS 51036, 51040 or 51100)

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.

Instructor(s): Martha Billingsley, Dries Kimpe Terms Offered: Autumn, Winter

Prerequisite(s): MPCS 50101 or programming waiver

Note(s): Non-MPCS students must receive approval from program prior to registering.

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).

Instructor(s): Anthony Nicholson Terms Offered: Spring

Prerequisite(s): MPCS 51040 or MPCS 51044 or instructor's consent

Note(s): Non-MPCS students must receive approval from program prior to registering.

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

Instructor(s): Sebastien Donadio Terms Offered: Spring

Prerequisite(s): Core Programming Requirement (MPCS 51036, 51040 or 51100)

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

Instructor(s): Jeffrey L. Cohen Terms Offered: Spring, Summer

Prerequisite(s): MPCS 50101 or programming waiver

Note(s): Non-MPCS students must receive approval from program prior to registering.

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.

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.

Instructor(s): Zachary Freeman Terms Offered: Autumn, Spring

Prerequisite(s): MPCS 50101 or programming waiver

Note(s): Non-MPCS students must receive approval from program prior to registering.

MPCS 53003. Advanced Databases. 100 Units.

The objective of this course will be to (i) expand the knowledge by covering new topics that represent the state-of-the-art in database management systems and distributed systems, and (ii) to build upon foundations developed in MPCS 53001 - Databases by covering topics in greater depth.

Instructor(s): Tanu Malik Terms Offered: Summer, Winter

Prerequisite(s): MPCS 53001 or Instructor's consent

Note(s): Non-MPCS students must receive approval from program prior to registering.

MPCS 53013. Big Data. 100 Units.

In this course, we will cover both the theory and practice of Big Data. To support practical experience with genuinely big data, we have arranged that all students will receive a substantial credit on the Google Cloud Platform courtesy of generous support from Google. To develop a sound understanding of the theory of Big Data, we will use Marz and Warren's Big Data textbook providing a conceptual architecture for Big Data systems. We will also cover important additional topics that invariably arise in real world applications of Big Data, such as like cleaning scraped data meant for human consumption to meet the needs Big Data systems. Students are required to bring a laptop to class every week.

Instructor(s): Michael Spertus Terms Offered: Autumn

Prerequisite(s): MPCS 50101 or programming waiver

Note(s): Non-MPCS students must receive approval from program prior to registering.

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

The course covers statistical methods for exploring, summarizing, and visualizing data sets, for modeling data using probability distributions, for making inferences about a population from samples, for testing hypotheses related to such inferences, and for describing relationships using linear and logistic regressions. It then examines in detail techniques from machine learning used for solving fundamental problems in data mining: classifying data through decision trees, nearest-neighbors, and Bayesian techniques; clustering data through k-means, hierarchical approaches, and density-based techniques; and performing association analysis through the Apriori algorithm. Students use Python for implementing algorithms 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, conditionals, 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.

MPCS 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.

Instructor(s): Amitabh Chaudhry Terms Offered: Spring

Prerequisite(s): B+ or above is required in each of the following classes:,,Core Programming requirement ,MPCS 55001 Algorithms ,MPCS 53110 Foundations of Computational Data Analysis..,If you are concurrently taking Algorithms with Machine Learning, a B+ or higher in MPCS 50103 Math for Computer Science: Discrete Math is required..,Programming in Python is necessary for the class. The following topics are required: use of lists, dictionaries, conditionals, classes, and file i/o...Students must have attended the Python workshop, have previous familiarity with these topics or be willing to teach themselves. Knowledge of this material will be expected.

MPCS 53112. Advanced Data Analytics. 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 Chaudhry

MPCS 53113. Natural Language Processing. 100 Units.

Can we predict how people will vote based on their twitter conversations? Can we identify 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 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.

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 we will focus on probabilistic and machine learning mechanisms that learn ambiguity resolution by training on large amounts of text corpora. These include sequence models such as Markov models, hidden Markov models, and conditional random fields. They also include classification and clustering techniques, such as logistic regression, naive Bayes, support vector machines, Gaussian mixture models, and EM clustering.

All through the course we will both im

Instructor(s): Amitabh Chaudhary 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): Non-MPCS students need to complete a course request form.

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): Anthony Nicholson Terms Offered: Winter

Prerequisite(s): MPCS 50101 or programming waiver

Note(s): Non-MPCS students must receive approval from program prior to registering.

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.

Instructor(s): Geradline Brady Terms Offered: Autumn, Spring

Prerequisite(s): MPCS 50101 and MPCS 50103

Note(s): Non-MPCS student must receive approval from program prior to registering.

MPCS 56420. Bioinformatics for Computer Scientists. 100 Units.

This course aims to introduce 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: Autumn

Prerequisite(s): MPCS 53001 and Core Programming requirements. Lectures and demonstrations will be conducted in Python. Python programming experience will be useful, but is not required as long as students are willing to dedicate sufficient time to obtain basic development and debugging skills in the language. The course is focused on developing solutions to biological problems, not on mastery of any particular language. Final projects will be implemented on Google Cloud Platform which supports Python, Java, PHP and Go.

Note(s): Non-MPCS students must receive approval from program prior to registering.

MPCS 56513. Digital Forensics. 100 Units.

In this course we will cover processes for investigations and evidence handling, types of evidence available, tools used in forensic investigations, recovery and preservation of data, and other forensic processes used in system incident response. We will use hands-on approaches with a number of tools and document results. Digital Forensics is a field of technology encompassing the investigation of digital devices as a part of incident response or data recovery. Forensic processes are used to recover evidence, determine the nature of an incident, puzzle together how the incident occurred and prepare evidence for potential court examination. In the Internet world of constant attacks, forensics have become an integral part of an incident response capability - to determine the nature of the attack, prepare evidence for further prosecution, if possible and to prevent future attacks. In this course we will cover processes for investigations and evidence handling, types of evidence available, tools used in forensic investigations, recovery and preservation of data, and other forensic processes used in system incident response. We will use hands-on approaches with a number of tools and document results. MPCS 52011 - Introduction to Computer Systems meets this prerequisite. Other core Systems courses may be used to meet this prerequisite with instructor's consent.

Instructor(s): Arlene Yetnikoff Terms Offered: Summer

Prerequisite(s): Good understanding of computer systems and architectures.

Note(s): Non-MPCS student must receive approval from program prior to registering.

MPCS 56515. Computer and Network Security. 100 Units.

The objective of this course is to provide a basic understanding of Information Technology security - and to build an understanding of the elements that should be in place for an IT environment to achieve an adequate security level. We will begin with a general overview of IT security and introduce a framework for addressing security needs across an enterprise. Major security objectives and technical mechanisms for attaining these objectives will be discussed, including cryptography, authentication systems, Public Key Infrastructure, and platform and network security mechanisms. This course will give an overview of the technical details involved in the platform and network levels of security, including hands-on usage of current tools used in the field. We will look at common TCP/IP applications and discuss their security vulnerabilities. The course material will be presented in a framework of understanding overall risks and how to address them. There will be a great deal of reading in this course. Students should have the ability to read and write in clear prose. Students in this course will be writing an in-depth paper or a project and should have the ability to write a substantial paper.

Instructor(s): Arlene Yetnikoff Terms Offered: Autumn

Prerequisite(s): Students in this course will use the Unix operating system as a basis of learning host security mechanisms and should have a basic familiarity with Unix as a prerequisite. Students should also be familiar with TCP/IP networks.

Students will be installing, configuring and running security tools obtained from the Internet as a part of their classwork.

Note(s): Non-MPCS students must receive approval from program prior to registering.

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.

Instructor(s): Andrew Siegel Terms Offered: Summer

Prerequisite(s): MPCS 50101 or programming waiver; MPCS 50103 or math waiver recommended., Languages: Required: familiarity with C/C++, Python, or Java (other language options are acceptable, but consult instructor first). Recommended: C and Python.



Font Notice

This document should contain certain fonts with restrictive licenses. For this draft, substitutions were made using less legally restrictive fonts. Specifically:

Times was used instead of Trajan.

Times was used instead of Palatino.

The editor may contact Leepfrog for a draft with the correct fonts in place.