C&! Course Descriptions

2026

Welcome to *C&!*!

We want you to have a *great* experience, learn good stuff, and make great friends for life. For that, it is important that you choose the right classes, ones which interest you and are at the right level. To accommodate the big age range at C&! we offer a wide range of classes. Read through the descriptions carefully and find classes that are right for you! We will make a great effort to get every student classes that are right for their level and interests. Class selection will start a few days after class description is made available.

Courses range from *Accessible*, that anyone at camp can enjoy, through *Intermediate* courses where some mathematical maturity is assumed (and may require some prerequisites), and all the way to *Prerequisites* courses that require more advanced prerequisites (typically linear algebra and/or calculus). The specific prerequisites depend on the course, and the details can be found in the description of the individual courses. Some courses fall under more than one category.

These divisions are meant to help you know what to expect, but remember they are imprecise! Also remember, just because a course is labeled *accessible* doesn't mean it is easy, and just because it is labeled *intermediate* doesn't mean it has no prerequisites! In every course we have the ability to take students as deeply into advanced mathematics as they want to go. Read the description carefully to find courses that are interesting for you and of the right level.

You'll find tags next to each course name. Here's what they mean.

H Hands-on course where you'll be making things and get out of your seat as part of learning

P Pencil and paper work will be an important part of learning in this course

L Course with significant lecture component

C Computer-based course, laptop needed

E Exploratory course where students are offered problems and staff support, but are expected to self-direct their work

Once students start registering, we will update some courses based on demand and send an updated list.

And now, to work! Click on the course name to see the full description:

Accessible	Intermediate	Prerequisites
How Languages Change: Indo-European Historical Linguistics (LP)		
How Languages Work: Sound & Syntax (LP)		
Intermediate Problem Solving (LPC)		
Fractals and Topology (LPC)		
Dome Math (HL)		
The Math of Video Games (H)		
Machines That Follow Rules (HP)		
Fast-growing Functions and Ridiculously Huge Numbers (LP)		
Mathematical Structures of Strategy (P)		
Numbers and Games (P)		
Puzzles, Paradoxes and Infinities (PE)		
Geometry Puzzles (PE)		
Probability Puzzles (LP)		
Graph Theory (LP)		
Induction (PL)		
Ducci Sequences (LP)		
Mathemagic (LP)		
Euclidean Algorithm (HP)		
Curved Worlds: Geometry on Surfaces (HP)		
Spherical Geometry (PL)		
	Computational Probability and Simulation (C)	
	Random Walks and the Long Run (HPL)	
	Talk with a Lisp (C)	
	The Geometry of Mechanical Speech (PL)	
	Hamiltonian Dynamics & Deterministic Chaos (PLC)	
	Geometry of Complex Numbers (LPC)	
	Quantum Mechanics (L)	
	Generating Functions (P)	
	Braid Groups and The Word Problem (LP)	
		Linear Algebra over Finite Fields (L)
		Seeing with Waves: Fourier Analysis and X-ray Vision (LC)
		Dissecting a Large Language Model (C)
		λ-calculus (LP)
		Quantum Computing (LPC)
		Design and Analysis of Secure Messaging Protocols (C)
		AI Security (LC)

How Languages Change: Indo-European Historical Linguistics (Todd Krause, all levels)

Do you speak English like Shakespeare? Do you even speak it like your parents or your teachers? Take some time to look at the differences and think about how those changes come about.

And did you know English is related to German? Look at some common words to check: Vater 'father', Mutter 'mother', zwei 'two', Hand 'hand'. But it's also related to Italian, Greek, Russian, and even Hindi. How does that work? What does it mean for languages to be "related"?

This class will look at how Historical Linguistics describes relationships among languages. We will mine English for clues as to how languages evolve over time. We will compare Old English to other ancient languages looking for similar patterns in both grammar and vocabulary: maybe Latin? Ancient Greek? Gothic? (Not just a way of dressing...) Sanskrit? Old Church Slavonic? Tocharian? (Is that actually a thing?...) Get ready to climb the family tree and learn some new — old — languages!

Open to students of all backgrounds and ages.

Style: Lecture (L), Pencil & Paper (P) (Back to top)

How Languages Work: Sound & Syntax (Todd Krause, all levels)

Consider: you have some crazy thought. By some process you can't explain in detail, you pass this thought into a jumble of sounds. Those sounds smash into somebody's ear and — poof! — suddenly that thought's in their head too! How in the world does that work?!? The trick is, the sounds aren't such a jumble after all. There's method to the madness: there's a system, and somehow we all carry around a copy of this system in our heads. Are all those copies the same? How much can they differ and still work? What kinds of systems are there in the world? What do they have in common?

This class will provide an introduction to the basics of modern linguistic analysis. What are nouns? You've seen them in English, but do all languages have nouns? Do verbs work the same way in all languages? In this class we will learn to analyze the fundamental components of a language: the different types of words that a language has. We will also explore how we identify the rules by which the elements fit together. What is the system we all carry around in our heads? The real challenge facing you as a budding linguist: how can you figure all this out for a language you don't even know yet?!? You might be surprised how some of the thinking you use in mathematics will keep you in good stead when analyzing languages. So come explore one of the great mysteries of the human mind: how it creates language!

Style: Lecture (L), Pencil & Paper (P) (Back to top)

Intermediate Problem Solving (Matthew Cho, all levels)

This course introduces a toolkit of mathematical thinking strategies through hands-on puzzles, games, and competitions. Each day focuses on a different problem-solving approach: we begin with logical deduction through constraint-based puzzles, then explore the art of mathematical proof, learning to navigate problems where the answer itself is uncertain. From there we turn to recursive thinking, estimation techniques, put to the test in a competitive estimathon, and expected value, where students learn to reason precisely about uncertainty. Throughout the week, the emphasis is on developing general strategies that transfer across mathematical contexts.

Style: LPC. (Back to top)

Fractals and Topology (Jeremy Van Horn Morris, all levels)

In this course we will learn the basics of the mathematical field of topology. We will explore interesting topological spaces and the fundamental objects that topologists study. Some foundational and very interesting examples will come from fractals and we will spend considerable time learning about how to design and make fractals, as well as computer-based tools for creating visualizations.

Prerequisites: Standard euclidean geometry and familiarity with cartesian coordinates. Some familiarity with transformations would be beneficial but will also be covered in the course.

Style: LPC. (Back to top)

Dome Math (Jason Cantrella, accessible)

A course with no prerequisites about geodesic domes. We'll learn about the mathematics of polyhedra and geodesic domes, build some scale models to get started, and then settle down to building a pair of Very Large cardboard domes: large enough to add some doors and windows and play in!

Style: LH. (Back to top)

The Math of Video Games (Hadas Zeilberger, accessible)

The way characters jump, spin, and move in your favorite video games is actually powered by a really cool type of math, called linear algebra!

In this class, we're going to peek behind the screen to see how it all works. We'll learn about "vectors", which are basically invisible arrows that tell the computer exactly where things should go. Then, we'll figure out how to stretch, shrink, and spin those arrows to make shapes move around.

The class will be filled with awesome demos and fun games so you can see exactly how this math transforms into the computer graphics you love.

Style: H. (Back to top)

Machines That Follow Rules (Aidan McCue, accessible)

Can a machine follow simple rules and still solve complex problems? In this course we'll explore how mathematicians and computer scientists design "machines" that perform tasks using only the simplest step-by-step instructions. This will lead us to one of the most famous ideas in computer science: the Turing machine. We'll experiment, play games, and even act out machines as a group to understand how these systems work, and to see what they can (and can't) do!

Style: HP. (Back to top)

Fast-growing Functions and Ridiculously Huge Numbers (David Metzler, accessible)

Many people are familiar with exponential functions such as 2^n and the factorial function n! and know that they grow very quickly—so 1000!, for example, is a very large number. How far can we take this idea? How large a (finite) number can we define precisely? We'll discover that pursuing this simple question leads to deep ideas about recursive functions and different flavors of infinity. Plus, you'll be able to destroy the competition in a "name the biggest number" contest!

Style: LP. (Back to top)

Mathematical Structures of Strategy (Wendy Tam, accessible)

This course investigates abstract systems of strategic interaction using tools from discrete mathematics, recursion, and symbolic reasoning. Students analyze how complex positions can be built from simple components, how values can encode advantage, and how structure governs optimal play. Along the way, the course highlights deep connections between strategy, number systems, and mathematical representation.

Style: P. (Back to top)

Numbers and Games (David Metzler, accessible)

How is a game like a number? How do you "add" two games to make a new one? How could you say that one game is "larger" or "smaller" than another? In this course we will explore some aspects of John Conway's theory of combinatorial games, and how they relate to his theory of surreal numbers. We will spend most of our time playing certain games that are designed to bring out the basic ideas of both theories, while always looking for patterns. We will see how numbers tell us about when games are won or lost, and how games tell us about the existence of new kinds of numbers, such as infinity minus one or one over infinity.

Style: P. (Back to top)

Puzzles, Paradoxes and Infinities (Vinay Kalyankar, accessible)

We will look at logic and other interesting puzzles, some paradoxes in mathematics, and we will spend some time in Hilbert's hotel which is always full and yet can accommodate infinitely many more guests. In the process, we will learn about different kinds of infinities.

Style: PE. (Back to top)

Geometry Puzzles (Vinay Kalyankar, accessible)

We will explore some puzzles in geometry that will require a basic understanding of high school geometry. The course will help in preparing for high school geometry for those students who have not yet taken it.

Style: PE. (Back to top)

Probability Puzzles (James Degnan, accessible)

We'll review basics of probability such as conditional probability, Bayes Theorem, and random variables, to look at some classic probability problems such as The Monty Hall problem, the St. Petersberg Paradox, Gambler's Ruin, and the probability that a stick broken in two places can have the three pieces form a triangle.

Style: LP. (Back to top)

Graph Theory (James Degnan, accessible)

This is an introduction to graph theory, where graphs are vertices with edges connecting them. Such graphs can be used to represent social networks, evolutionary trees, and security cameras in a

building. We will cover topics such as Eulerian cycles (with an application to the Bridges of Königsberg), Hamiltonian cycles, and the traveling salesperson problem.

Style: LP. (Back to top)

Induction (Vivian Fang, accessible)

Let's learn how to prove statements for (countable) infinities, without doing infinite work! Concepts: proofs by induction, strong induction. We will then go through some neat proofs in computer science that use induction: stable marriage, graph algorithms, and if we have time, balls-into-bins.

Style: LP. (Back to top)

Ducci Sequences (Cornelia Van Cott, accessible/intermediate)

Draw a square and write an integer at each corner. Now we are ready to create a Ducci sequence, as follows: (1) Find the positive difference of the integers at adjacent corners. (2) Create a new square with the resulting four numbers. (3) Repeat this process. What's going to happen? How does the Ducci sequence's behavior depend on the numbers we started with? What if we start with a pentagon or triangle instead of a square? These sequences were first introduced by Italian mathematician E. Ducci in the 1800s. The problem resurfaced in multiple contexts and is still studied today. In this class, we will fully investigate the behavior of Ducci sequences and find several surprises along the way.

Prerequisites: Experience with basic algebra, including functions and graphing points and lines on the plane.

Style: LP. (Back to top)

Mathemagic (Cornelia Van Cott, accessible)

In this class, we'll study the mathematics behind card shuffling and examine several mathematical card tricks. No card-shuffling experience is required!

Style: LP. (Back to top)

Euclidean Algorithm (Rolfe Schmidt, accessible)

In this class we will look at one of the most fundamental algorithms in Mathematics: Euclid's algorithm for computing the Greatest Common Denominator (GCD) of two natural numbers. If you have seen GCDs before, you may be wondering why computing them is so important. If you have those doubts, then this class will change your mind! We will start by using it to solve Diophantine equations (algebraic equations over the integers) and then see how these solutions, seen through a modern lens, let us perform the basic operations that underlie much of modern cryptography.

Style: HP. (Back to top)

Curved Worlds: Geometry on Surfaces (Gennady Uraltsev, accessible/imtermediate)

What is a straight line? Do the angles of every triangle add up to 180°? Can you walk straight forever and end up back where you started?

These questions may sound like puzzles, but they come up in physics, where space itself can be curved by massive objects like stars and black holes. What does it mean for space to be curved? To understand this, we need to rethink what space is and how geometry works within it.

In this course, we explore geometry on surfaces that are not flat. By cutting, gluing, and counting triangles, we will experiment with curved surfaces that live in our three-dimensional world. From the outside, we can see their full shape, but what would the world look like to something living on the surface?

Imagine an ant crawling along such a surface. It can measure distances and angles, but it may have no sense of "up" or "down." What kind of geometry would the ant discover? How would it define straight lines, triangles, or curvature? And what can that perspective tell us about our own universe?

Geometry is not about how things look from the outside. It is about what can be measured from within.

Expect hands-on activities, drawing, building, and exploration as we develop new geometric intuition from the ground up.

Familiarity with basic geometry (triangles, angles, parallel lines) is assumed. (Back to top)

Style: HP. (Back to top)

Spherical Geometry (Jason Cantrella, accessible/intermediate)

This course is designed for students who have had a high-school(ish) level course in plane geometry and know some trigonometry. What does geometry become on the sphere? We'll study some essentials of classical spherical geometry, learning how sailors and astronauts navigate on the globe by solving spherical triangles, and have some time for problems and solutions, leading up to Maehara and Martini's solution to Newton's 13 ball problem: how many disjoint (unit) spheres can simultaneously touch one unit sphere?

Style: LP. (Back to top)

Computational Probability and Simulation (Wendy Tam, intermediate)

This course studies probability through programming, simulation, and computational experimentation. Students write programs to model random events and investigate how stochastic systems unfold over time. Topics may include random walks, Markov chains, Poisson processes, Monte Carlo methods, and other stochastic processes, with emphasis on using computation to build intuition, test ideas, and analyze probabilistic behavior.

Prerequisites: proficiency coding in python or R.

Style: C (Back to top)

Random Walks and the Long Run (Tanay Mehta, accessible/intermediate)

If you wander randomly through a city, hop randomly between web pages, or roll your way around a Monopoly board — where do you end up? Strangely often, the answer doesn't depend on where you started. Random processes have a way of forgetting their past and settling into patterns you can predict. So how do we describe these processes mathematically? Why do they converge, and how fast? And how does this same idea power Google's original search ranking, predict your next text message, and tell us how many shuffles it takes to randomize a deck of cards? Spend the week exploring random walks and Markov chains, and learn what the long run really looks like.

Style: HPL (Back to top)

Talk with a Lisp (Daniel Hadar, intermediate)

In this course we will be exploring Lisp, a beautifully expressive programming language where the source code of a program is no different than the data it modifies. Being purely functional, Lisp is very closely related to the λ-calculus and offers a unique mathematical perspective to solving problems and manipulating data. Typical programming notions like looping and classes are replaced entirely by clever uses of recursion, and the language is flexible enough for programs to modify their own syntax as they run.

Lisp is dramatically different from languages like Python or C++, so students new to programming will not be at any disadvantage.

Style: C

(Back to top)

The Geometry of Mechanical Speech (Todd Krause, intermediate/prerequisites)

It turns out computers can produce human language just about as well as humans! The Large Language Models (LLMs) that produce this speech are based on artificial neural networks, which act as crude models of how the human brain works. But in the end, an LLM works with pure math, not language. What is it really doing with those numbers to mimic something as complex as human thought?

This class bridges the study of linguistics and mathematics through a tour of the fundamental mathematical topics underlying modern LLM architectures and Deep Learning in general. At each step, we will focus on visualizing exactly what the math is doing. Sure, LLMs apply matrices to vectors, but can you picture what that transformation actually represents in space? How can we visualize general functions as neural networks? Are these machines really applying the high-level concepts from differential geometry? As we survey these topics, we'll constantly compare them with our understanding of human language and linguistics to see what LLMs can teach us about the human brain and its language capacity.

No prior linguistic knowledge is assumed. Some knowledge of calculus, including derivative rules and integration, will be useful. You should be comfortable with basic manipulations of vectors and matrices; prior exposure to linear algebra will be helpful, though not required.

Style: PL

(Back to top)

Hamiltonian Dynamics & Deterministic Chaos (Todd Krause, intermediate/prerequisites)

Chaotic is not the same as random. It turns out that Newton's laws of motion hide far more complexity within them than you might've been led to believe. For centuries, classical mechanics left the impression of a Clockwork Universe, where everything ticks along like the intricately enmeshed, predictably advancing gears of a finely wrought timepiece. But with the advent of advanced techniques of nonlinear analysis and computer modeling, we have found that within this rigid determinism lie the seeds of chaos: systems can display completely unpredictable behavior, even while obeying iron-clad dynamical laws.

This class examines classical mechanics from the broader perspective of dynamical systems, exploring the interplay between mathematical theory and numerical experiment. We'll adopt the Hamiltonian perspective and explore the powerful visualization techniques it affords. We'll then make our way to unstable manifolds and homoclinic tangles (yes, it's a thing!). By constructing surfaces-of-section, we'll explore how the Golden Mean explains why it seems fusion reactors are only 20 years away... and always will be!

Prerequisites: no prior physics knowledge is assumed. You should be comfortable with basic manipulations of vectors and matrices; prior exposure to linear algebra will be helpful, though not required. You should also be comfortable with basic calculus, ideally including the convergence of series. Some of our

explorations will feature a computational component, but you do not need prior programming experience to benefit from them.

Style: PLC (Back to top)

Geometry of Complex Numbers (Jeremy Van Horn Morris, intermediate/prerequisites)

Complex numbers are integral to many advanced areas of computation, physics, and modeling, and their arithmetic has a much more interesting (and rigid) structure than real numbers. In this class we will learn the fundamentals of the complex numbers, their arithmetic, ways of understanding them, and the amazing geometry that they demonstrate.

Prerequisites: Trig functions and trig identities, basic polynomial algebra, unit circle and polar coordinates, parametric equations.

Style: PLC (Back to top)

Quantum Mechanics (Julien Piet, Intermediate/prerequisites)

This course provides an introduction to the fundamental principles of quantum mechanics and their applications. Students will learn how matter and energy behave at the atomic and subatomic scale, where classical physics no longer applies. We'll explore concepts such as wave-particle duality, quantum superposition, the uncertainty principle, and quantum entanglement. Notions of algebra are helpful. No background in physics is necessary.

Style: L (Back to top)

Generating Functions (Matthew Cho, Intermediate/Prerequisites)

This course introduces generating functions, a powerful tool in counting and probability. We'll start with the definition of generating functions, and apply them to problems that are difficult to approach directly. We will apply generating functions to solve recurrence relations, including the explicit formula for the Fibonacci numbers, and end by using generating functions to prove facts about partitions of integers.

This course assumes familiarity with algebra and knowing what a binomial is, but there will be problems hard enough to challenge even the strongest students.

Prerequisites: A strong command of algebra (for instance having done AOPS intro to algebra) is important. One should also have some background in counting, for instance, knowing what a binomial is and how to compute one.

Style: P (Back to top)

Braid Groups and The Word Problem (Aidan McCue, Intermediate/Prerequisites)

Given two descriptions of a mathematical object, how can we tell whether or not we have described the same thing? In this course we'll explore braids, topological objects made by strands weaving over and under each other. We'll learn how to work with these objects using algebra, describing them with "words", strings of symbols that record how the strands cross. Then comes the real challenge: when do two different looking words actually describe the same braid? By the end of this course, you'll learn an algorithm that can always answer this question. Along the way, we'll mix visual thinking and abstract ideas to uncover the hidden structure behind braids.

Prerequisites: Familiarity with groups, and in particular permutation groups.

Style: LP (Back to top)

Linear Algebra over Finite Fields (Hadas Zeilberger, prerequisites)

Dive deep into the mechanics of linear algebra over finite fields through rigorous, proof-based problem-solving. We will build a strong foundation with vector spaces and matrices before tackling advanced concepts like eigenvectors and annihilators. These techniques are essential for many applications, including elliptic curve cryptography and post-quantum cryptography.

Prerequisites: Linear Algebra (matrix multiplication and inversion, determinants), and basic theory of the finite fields $\mathbb{Z}/(p)$.

Style: L (Back to top)

: Seeing with Waves: Fourier Analysis and X-ray Vision (Gennady Uraltsev, Prerequisites)

What does the function $\sin(x^2)$ sound like? Can you hear whether a graph has corners, or whether it can be drawn without lifting your pen? Can you hear the shape of a drum? Can you determine what an object looks like just from its X-rays?

In this course, we explore how complicated signals can be broken down into simple waves. This is Fourier analysis, where functions are sums of pure oscillations. We will learn how to extract these components and how to put them back together. We use this to efficiently encode, compress, and transmit signals.

Video by 3Blue1Brown: But what is a Fourier series?. If the link doesn't work use this: `https://www.youtube.com/watch?v=r6sGWTCMz2k`

In many real situations, we do not get to choose what we measure. What if we only measure the strength of signals passing through an object from many different directions, like an X-ray machine does? Fourier analysis can be used to reconstruct what the object looks like. This is an example of an inverse problem: starting from measurements and we try to figure out what caused them. Fourier analysis can help us here as well. However, we will also see that not all inverse problems can be easily solved.

Expect computer-aided computation, visual exploration, and conceptual thinking. We will use simple computer tools for simulations and experiments, but no prior programming experience is required.

Prerequisites: Algebra and basic functions, trigonometric functions, and basic calculus (integrals). Familiarity with complex numbers and basic linear algebra will be helpful.

Style: LC (Back to top)

Dissecting a Large Language Model (Daniel Hadar, Prerequisites)

In this course we will be poking around at the internals of a large language model (GPT-2) to see how it works. We'll explore tokenizers, attention heads, and feed-forward network layers and see how they work in a real model.

Prerequisites: A familiarity with programming fundamentals in a language like Python will be important for following along with the material.

It will also be important to have a background with concepts like gradients from calculus and a general comfort with linear algebra.

Style: C (Back to top)

λ Calculus (Vivian Fang, Prerequisites)

We'll show how any computation can be expressed by a function, and dig into what "undecidability" means. We'll cover concepts including the Church-Turing thesis, the halting problem, Gödel's incompleteness theorem, and Rice's theorem. See a related course.

Style: LPC (Back to top)

Quantum Computing (Tanay Mehta, prerequisites)

Quantum computers exploit the peculiar nature of quantum mechanics to perform calculations in a novel way that is sometimes more powerful than classical computers. In this course, we'll learn the mathematical basis for manipulating "qubits" and understand how they behave differently from traditional "bits." We'll build up the core machinery of quantum algorithms — superposition, entanglement, quantum circuits, the Quantum Fourier Transform, and the Phase Estimation Algorithm — and program everything ourselves on a simulator called qiskit. We will program in python 3 (no experience is needed). If we make good time, we'll close the week by assembling these pieces into Shor's algorithm, the celebrated polynomial-time quantum algorithm for factoring integers that would, if run on a large enough quantum computer, break the cryptography securing most of the internet today. Students need to bring a laptop (any laptop will do).

Students that took Quantum Computing last year and encouraged to take this course. The review will be very valuable and the topics are not the same.

Prerequisites: Probability theory, linear algebra (matrices, determinants, etc).

Style: LPC (Back to top)

Design and Analysis of Secure Messaging Protocols (Rolfe Schmidt, prerequisites)

We will go through a progression of secure messaging protocols, analyzing the security properties of each early candidate with the Tamarin Prover - a formal verification tool - until finally arriving at Signal's currently deployed Triple Ratchet and PQXDH protocols.

Prerequisites: We will go through a progression of secure messaging protocols, analyzing the security properties of each early candidate with the Tamarin Prover - a formal verification tool - until finally arriving at Signal's currently deployed Triple Ratchet and PQXDH protocols.

Style: C (Back to top)

AI Security (Julien Piet, prerequisites)

This course provides an introduction to the security of the deep learning algorithms used in modern AI application. We will study the fundamental building blocks of deep learning, learn about how AI models are trained, and show how the same tools can be used to derail models to produce bad outputs. We will apply this to two important applications: image classifiers, and language models.

Prerequisites: Notions of calculus (derivatives) and linear algebra are helpful.

Style: LC (Back to top)