# CLASS DESCRIPTIONS—WEEK 1, MATHCAMP 2007

## Contents



## Classrooms

Our classrooms are in the following buildings:

- Diamond (building 19 on your campus map)—rooms D241 and D243
- Lovejoy (building 7)—rooms L212 and L213
- Keyes (building 4a)—rooms K102 and K103
- Arey (between buildings 4 and 4a)—lecture hall A005 (in the basement).

Recall that our dorms are building 13 and the cafeteria is building 1 (Dana). Our computer lab is also in Diamond.

## 9am Classes

**Knot theory.** (∗∗, Noah, week 1 of 1)

The theory of knots in 3-dimensional space is a particularly exciting subject because although many of the key advancements have been made only in the last 30 years, these ideas are nonetheless accessible to everyone. This class will begin by examining knots in their purest form: actual pieces of string. Eventually, we will learn how to describe knots with a drawing that encodes a finite amount of data, and discuss the fundamental question of how to tell when two drawings describe the same knot. Next, we will discuss the reverse question of how to distinguish two knots. The key tool here will be constructing knot invariants.

*Prerequisites:* None.

*Homework:* Recommended.

*Related to:* Low-dimensional topology (Week 2); Algebraic topology (Week 4)

**Generating functions, Catalan numbers, and partitions.** (∗∗–∗∗∗, Mark, week 1 of 1)

Generating functions provide a powerful technique, used by Euler and many later mathematicians, to analyze sequences of numbers; often, they also provide the pleasure of working with infinite series without having to worry about convergence.

The sequence of Catalan numbers, which starts off 1, 2, 5, 14, 42, . . . , comes up in the solution of many counting problems, involving voting, lattice paths, polygons, . . . . We'll use generating functions to come up with an explicit formula for them.

A *partition* of a positive integer $n$ is a way to write $n$ as a sum of one or more positive integers, say in nonincreasing order; for example, the seven partitions of 5 are

$$5,\ 4+1,\ 3+2,\ 3+1+1,\ 2+2+1,\ 2+1+1+1,\ 1+1+1+1+1.$$

The number of such partitions is given by the partition function $p(n)$ ; for example, $p(5) = 7$. Although an "explicit" formula for $p(n)$ is known and we may even show it, it's quite complicated. In our class, we'll combine generating functions and a famous combinatorial argument due to Franklin to find a beautiful recurrence relation for the (rapidly growing) partition function. This formula was used by MacMahon to make a table of values for $p(n)$ through $p(200) = 3972999029388$, well before the advent of computers!

*Prerequisites:* Summation notation; geometric series. Calculus is useful. Some experience with more general power series may help, but is not really needed.

*Homework:* Recommended

*Related to:* Other combinatorics

**Information and coding theory.** (∗∗∗, Mira, week 1 of 3)

Imagine you're trying to transmit a string of bits (0's and 1's) over a bad connection, in which each 0 has 10% probability of being flipped as a 1, and vice versa. What can you do to make sure that your message is received correctly?

One possible solution is to transmit each bit multiple times. For instance, if you wish to send the message 10, you transmit 111 000. Suppose I receive 111 001. Then I know that an error has occurred—most likely the sixth bit has flipped from 0 to 1—so I decode your origininal message correctly. Of course, if two errors had occurred (so that I received 111 101), I would have decoded your message incorrectly as 11. But two errors are unlikely, so the overall probability of error has clearly gone down. (You can compute that it's now 2.8% per bit intead of 10%.) We could decrease it still further by transmitting each message bit five times, or seven times, or more. (Why are we using odd numbers here?) This is a very rudimentary example of an *error correcting code.*

There's an obvious price: when you send each bit three times, the *rate of transmission* goes down. You're now communicating only 1 bit of original message per three bits transmitted, so we say that the rate of this code is $\frac{1}{3}$. If you transmit with even more repetition, you can get the probability of error to be as small as you want—but the rate of transmission will go to zero as well! You can't get something for nothing.

Except that, amazingly, you can!! In 1948, Claude Shannon pioneered the subject of information theory by proving a remarkable theorem: every noisy communication channel has a *capacity*—a maximal rate at which you can transmit information virtually error-free. Below this rate, you can get the probability of error to be as close to zero as you want; above this rate, it's hopeless. For instance, the capacity of the channel in our problem turns out to be about 0.53. So Shannon's theorem says that, at least in theory, you and I can devise a code in which you transmit only twice as many bits as your original message contains (rate $= \frac{1}{2} < 0.53$), and I can recover each bit of your message with probability of error $\frac{1}{\text{googol}}$ (or $\frac{1}{\text{googolplex}}$, or whatever we agree we can tolerate).

Ponder that for a second: it's mindboggling!! In class, we'll see the proof of this theorem, and you'll probably think that it's more mindboggling still.

The catch in the previous paragraph is in the words "at least in theory": that's where information theory ends and coding theory begins. In the first 50 years after Shannon's theorem, many clever error-correcting codes were developed and used successfully, in everything from CD players (that's why a scratch on your CD doesn't ruin the music) to NASA's communication with its Jupiter-orbiting spacecraft Galileo. These codes also led to the development of some fascinating mathematics. However, none of them came even remotely close to the promise of Shannon's theorem—and there's a good reason for this, which we will discuss in class. Then, in the late 1990's, some people started thinking about coding theory in a totally new light, using techniques from probabilistic modeling, graph theory, and dynamic programming instead of abstract algebra. This new perspective on codes, which has led to incredible progress in the last decade, is what we will focus on in the second half of the course.

*A note to MC06 alumni:* This class will have a different emphasis from my information theory class last summer: the goal this year is to get to the Noisy Channel Coding Theorem as soon as possible, prove it, and then talk about actual codes. If you took my information theory class at MC06, there will be some review, but I don't think you'll be bored. Depending on how much you remember, you *might* get away with skipping the first week—but talk to me first!

*Prerequisites:* Logarithms, basic probability.

*Homework:* Required.

*Related to:* Cryptography (Weeks 2–4); Zero-knowledge proofs (Week 3)

**From Greece to Galois: Constructiblity of regular polygons (Moore method).** (∗∗∗∗, Alfonso, week 1 of 4)

What do the following two problems have in common?

- Which regular polygons can be constructed with a ruler and a compass? (I won't reveal the nice pattern yet, but I can tell you that a regular heptagon cannot, whereas a regular 17-gon can.)
- Which polynomial equations can be solved with radicals (i.e., with a method similar to the quadratic formula that you know and love)? (Any quartic equation can be solved in a similar manner, but $x^5 + 3x + 6$ cannot).

Both problem are very easy to state (any eighth grader could understand the statement) but their answers require Galois theory (which is sometimes learned in grad school). Galois theory is a very powerful and beautiful area that lies in the intersection of many branches of pure mathematics: rings and fields, linear algebra, group theory, number theory, polynomials and factorization; and which I believe deserves to be more widely known. Also, impossibility proofs are among my favourites!

This course will be aimed at solving the problem of constructibility of polygons. We will start with all the necessary abstract algebra, move towards the Fundamental Theorem of Galois Theory, and then finally apply it to solve the constructibility problem. The highest prime $p$ for which I have actually constructed a regular $p$-gon was 17 (as a special present built for a special person on a fireworks night—I am not kidding). I hope one of you will dare to build a regular 257-gon.

If there is time at the end, we will attack the harder problem of solvability of polynomial equations with radicals. Otherwise, I will offer it for Week 5.

*Prerequisites:*

- arithmetic (up to every integer can be written as a product of primes in a unique way) and modular arithmetic,
- equivalence relations,

- trigonometry,
- complex numbers (I will offer a crash course sometime during Week 1),
- linear algebra (if you are not comfortable with the statement "every basis of a finite dimensional vector space has the same cardinality", then you should take "Linear algebra" in Weeks 1–2),
- group theory (if you are not comfortable with the statement "the order of a subgroup divides the order of the group", then you should take at least the first week of "Groups and symmetry" in Week 1. Also, if you are unfamiliar with actions, you should take "Group actions" in Week 3.).

If you want to take this class but are missing some of the prerequisites, talk to me.

*Homework:* This course is Moore Method. In other words: you will do all the work. You will receive handouts with definitions, motivation, and a list of theorems and exercises, but without a single proof. You are expected to work daily on preparing those proofs, which you will then take turns presenting in class. The class will then discuss the proofs together until we get a completely rigorous proof for every statement, with (hopefully) minimal help from me. Group work is encouraged, and I will be always available for help at TAU, but you are forbidden from consulting any book. I want to allow you the pleasure of "discovering" all the results on your own, even if other people have done it before.

*Related to:* Constructible or not? (Week 4)

## 10am Classes

**Introductory combinatorics.** (∗, M@, week 1 of 1)

If you don't know what factorials, binomial coefficients, combinations, and permutations are, you should probably take this class. We will discuss the basics of combinatorics, particularly basic ways of counting things, and also the Binomial Theorem.

Sample problems:

(1) How many different ways are there of putting 15 balls in 4 boxes?
(2) How many different ways are there to play 9 red bricks and 5 white bricks in a row? What if no two white bricks can be next to each other?
(3) What is the coefficient of $x^4$ in the expansion of $(x + 1/x)^{10}$?

*Prerequisites:* None.

*Homework:* Required.

*Required for:* All sorts of mathematics.

*Related to:* Generating functions, Catalan numbers, and partitions (Week 1); Enumeration celebration (Week 3); The Probabilistic Method (Week 4)

**Point-set topology.** (∗∗, Nina and Yvonne, week 1 of 2)

What does continuity of a function actually mean? Intuitively it means nearby points are mapped to nearby points. The field of point set topology comes into play when we change our definition of what it means to be "close together". One might think that the identity function on the real numbers, $f(x) = x$ is a continuous function, but this actually depends on the topology we put on the line. We can consider extremely unintuitive topologies on the real numbers. For example, in some topologies on the real numbers, contrary to our normal intuition, we can't find non-intersecting "intervals" around 0 and 1.

This course introduces fundamental definitions: topologies, open and closed sets, continuous maps, and homeomorphisms. These are the first building blocks in the fields of differential geometry, differential topology, and algebraic topology, among others.

*Prerequisites:* None.

*Homework:* Required.

*Required for:* Algebraic topology (Week 4; metric spaces can substitute)

*Related to:* Low-dimensional topology (Week 2); Orientability (Week 2); Metric spaces (Week 4)

**Groups and symmetry.** ($**$, Leigh, week 1 of 2)

Week one of Groups and Symmetry is "the beginning of a beautiful friendship". You've probably already seen quite a few mathematical groups, such as integers ($\mathbb{Z}$), complex numbers ($\mathbb{C}$), modular integers ($\mathbb{Z}_n$), and permutations ($S_n$), to name a few. These and other examples will pop up as we get to know the basic tools of group theory. And we'll explore how some of these groups describe a phenomenon which has inspired people from the ancient Greeks to today's artists (know a guy named Escher?)—symmetry. In particular, we'll play with models of the regular polygons and polyhedra, working up to the point where we can find their symmetry groups.

Examples, rather than theorems, will be the theme of the first week. Even if you already know something about groups, you might want to pop in for the symmetry fun. And group theory will come up in many other courses, so this is a good chance to get a taste of what's to come.

*Prerequisites:* None.

*Homework:* Recommended.

*Required for:* Many things! This class is specifically a prerequisite for Group actions (Week 3), Wallpaper patterns (Week 4), and Rubik's cube and other puzzles (Week 4).

However, *some* exposure to group theory is also required for the following classes; contact the instructor to find out how much: Cryptography (Weeks 2–4), Greece to Galois (Weeks 1–4), Representation theory (Week 3), Geometric group theory (Week 3), Musical orbifolds (Week 4), Problem solving with abstract algebra (Week 4), Algebraic topology (Week 4), Klein's quartic (Week 5).

**Real analysis.** ($***$, Ari, week 1 of 3)

You think about real numbers all the time, but do you actually know what they are? This course will be a rigorous exploration of the properties of the real numbers, sequences, and functions. We will construct the real numbers from scratch, then discuss several notions of convergence and continuity. The behavior of these creatures may shock you: series which can converge to any limit, depending on the order of summation! One-dimensional curves that fill n-dimensional space! Continuous functions which are nowhere differentiable! By the end of this course, you will be unafraid of epsilons.

*Prerequisites:* None.

*Homework:* Required.

*Related to:* Multivariable calculus (Week 1); Complex analysis (Weeks 2–3); Divergent series (Week 3); Measure theory (Week 4); Generalized Riemann integration (Week 4)

**Algebraic Geometry.** (∗∗∗∗, Dave J, week 1 of 2)

Take a polynomial and graph it. Your graph is an example of an affine variety. In a high school class like algebra or calculus, you may have learned that certain geometric properties of your graph, such as extrema, inflection points, and asymptotes, can be determined directly from algebraic properties of your polynomial. More generally, we could study the set of solutions to a system of any number of polynomials in any number of variables in any degree. Given such an object, we might ask which of its geometric features can be determined from a purely algebraic perspective.

In this course, we will study the rich and beautiful interplay between algebra and geometry, and we'll see that these questions naturally lead us to the theory of algebraic structures called rings. We will examine many different properties of rings, and what they indicate geometrically.

*Prerequisites:* None.

*Homework:* Required.

*Related to:* Noncommutative ring theory (Week 1); From Greece to Galois (Weeks 1–4); Category theory (Week 4)

*Required for:* Klein's quartic (Week 5)

## 11am Classes

**Linear Algebra.** (∗∗, Mira and Marisa, Week 1 of 2)

Linear algebra is the area of math that deals with vectors and matrices. It is one of the most useful methods in mathematics, both within pure math and in its applications to the real world. One could argue that most of what mathematicians (and physicists, and engineers, and economists) do with their time is try to reduce hopelessly complicated non-linear problems to linear ones that can actually be solved. Thus for many applied fields, the most important math to know is not calculus, but linear algebra.

We're going to start out on the plane, where linear algebra springs out of geometry. We'll define linear maps and give an intuitive preview of one of the central themes of linear algebra—eigenvectors and their eigenvalues—which we will encounter in much more generality in Week 2. Then we'll leave our two-dimensional pictures behind and introduce the more general concepts of vector space, linear independence, and basis—culminating in a purely algebraic definition of the concept of dimension. (If all this sounds familiar, skip the first week!)

In the second week, we'll talk about inner products, orthonormal bases, and diagonalization, leading up to a big theorem about eigenvectors of symmetric matrices: the Spectral Theorem. If time permits, we'll look at some cool applications of the Spectral Theorem taken from genetics and/or image processing. More applications can be found in the "Mathematics of Google" class that Mira will teach in Week 4.

*Prerequisites:* None.

*Homework:* Required, daily.

*Required for:* Greece to Galois (Weeks 1-4), Quantum mechanics (Week 2), Geometric graph theory (Week 2; recommended), Hyperplane arrangements (Week 3), Representation theory (Week 3), The Mathematics of Google (Week 4), The Rubik's cube and other puzzles (Week 4), Quaternions (Week 5)

**Introductory number theory.** (∗∗, David, Noah, week 1 of 2)

Choose your own adventure in elementary number theory!

(Page 1) You find yourself in the woods in Maine. You discover a description of elementary number theory on the Fundamental Toolkit list.
- You think "I already know all of this stuff!" (Go to Page 7)
- You think "Wow, I better not leave Mathcamp without learning all of this!" (Go to Page 3)

(Page 2) Welcome to Cyborg Elementary Number Theory with David Roe. Spend two weeks using computers to explore the basic ideas and more advanced topics in elementary number theory. We will learn about greatest common divisors, modular arithmetic, primes and primality testing, factorization and the Chinese remainder theorem. We will have the opportunity to program interesting algorithms related to these topics, including a longer mini-project in computational number theory. This course will also serve as a two week introduction to programming in Python, an easy to learn programming language, and SAGE, an exciting open source computer algebra system.

(Page 3) An alien lands and demands that you tell her whether 4294967297 is prime.
- You feel a deep urge to find a computer and implement a primality testing algorithm. (Go to Page 2)
- You grab your calculator and divide by 2, then by 3, then by 4, then by 5... (Go to Page 6)
- You think, there must be some way to do this using only a paper and pencil! (Go to Page 4)

(Page 4) Welcome to Luddite Elementary Number Theory with Noah Snyder. This is a *one* week course; during the days when the cyborg class is learning to use computers, we will be pushing the limits of what can be done with paper and pencil. Our motto will be what would Euler do? We will be covering greatest common divisors, modular arithmetic, primes, Fermat's little theorem, and Fermat numbers.

(Page 5) Welcome to the Secret Elementary Number Theory. In this class we will be proving Fermat's last theorem.

(Page 6) You divide by $m$, then by $m + 1$, then by $m + 2$, ... (Go to Page 6)

(Page 7)
- You want to learn about SAGE, the computer algebra system. (Go to Page 2)
- You feel no particular desire to play with computers. (Go to another class)

*Prerequisites:* None.

*Homework:* Required.

*Required for:* All sorts of things, especially other number theory.

*Related to:* Quadratic reciprocity (Week 2); Multiplicative functions (Week 2); Unique factorization (Weeks 2–3); Diophantine approximation (Week 4)

**John Conway.** (Usually ∗∗∗–∗∗∗∗)

**Multivariable calculus (crash course).** (∗∗∗, Mark, week 1 of 1)

In real life, most interesting quantities depend on several variables (such as the coordinates of a location, the time, the temperature, etc.). As a result, ordinary (single-variable) calculus isn't enough to solve most problems. This class will quickly take you through the basics of calculus of several variables. As time permits, we'll see some cool applications in and outside math, for instance:

- if you're in the desert, in what direction should you go to cool off as soon as possible?

- how large is the total area under a bell curve?
- what force fields are consistent with energy conservation?

With luck, we'll also cover Green's Theorem, which will be used next week in the complex analysis course.

*Prerequisites:* Single-variable calculus.

*Homework:* Recommended.

*Related to:* Real analysis (Weeks 1–3)

*Required for:* Complex analysis (Weeks 2–3)

**Methods of Attack on P vs. NP.** (∗∗∗∗, Dan, week 1 of 1)

This will be a continuation of the "Theoretical Computer Science" class from last year (or from 2005).

One of the most promising attacks on the P vs. NP question has been via the notion of *circuits*, which present a new model of computation different from (but related to) Turing machines. Instead of proving bounds on running time (which is hard), we can prove bounds on circuit size, which might be easier. Indeed, there are some bounds on circuit size that seem to come tantalizingly close to proving $\mathsf{P} \neq \mathsf{NP}$—but still fall short. The first half of this class is going to be an introduction to circuits and the methods they present for attacking lower-bound problems in complexity theory. The second half is going to use circuits to discuss the class EXPTIME of exponential-time problems, the class NEXPTIME, and how the relationship between those two classes might affect the relationship between P and NP. We'll understand what problems that take an exponential amount of computation look like, and even what it means to be NEXPTIME-complete. The result is beautiful and unexpectedly simple.

Be prepared for a fast-paced review of material from last year followed by much more cool TCS about current approaches to research in theoretical computer science.

*Prerequisites:* TCS from Mathcamp 2006 or 2005.

*Homework:* Recommended.

*Related to:* Computable functions (Weeks 3–4)

## 1:10pm Classes

**Proof Techniques.** (∗, Dan, week 1 of 1)

Learn the black art of proof techniques, where mathematicians seem to finish strange and difficult proofs with ease. Understand what it means when your professor says "this follows by induction," or "pigeonhole applies," or why they might shout "contradiction!" excitedly upon the conclusion of a proof. (Shouldn't that be a *bad* thing?)

We're going to study methods of proof and problem solving using a collection of interesting, unusual, *fun* problems (and we'll prove that $\sqrt{2}$ is irrational in three different ways)! Join us if you want to master these various proof techniques!

*Prerequisites:* None.

*Homework:* Required.

*Required for:* Everything!

**Combinatorial Game Theory.** (∗∗, Alfonso, week 1 of 1)

Consider the following variation of Problem 9 on the Qualifying Quiz. We start with a deck of $n$ cards, and mix them so that some are face up and some are face down. Lay them out in a straight line. A turn in this game consists of removing a face-up card and flipping over its immediate neighbors, if any. (Two cards are not considered immediate neighbors if there used to be a card between them that is now gone.) You and I take turns in moving. The first player who cannot longer move (because there are no face-up cards left) loses. If the starting configuration is UDUDUDUD, do you prefer to go first or second?

In Combinatorial Game Theory we like to study games like the one above: games with complete information, where no luck is involved. We do not like to play them so much as to analyze them, finding a winning strategy, and proving why it works. I will tell you some of the main techniques that allow us to attack a wide variety of such games.

*Prerequisites:* None. Do not come to this class if you already took it last year!

*Homework:* None, but I will offer some projects based on this course for whoever is interested.

**Zoology of polytopes.** (∗∗∗, M@, week 1 of 1)

Polytopes are the $n$-dimensional version of polygons (2-polytopes) and polyhedra (3-polytopes). We will investiage some basic properties of polytopes in this course, emphasizing special families of polytopes (regular, simplicial, simple, etc.), and with special attention to the $f$-vector.

The $f$-vector records the number of faces of each dimension for a polytope. For example, the cube has 8 *vertices* (0-dimensional faces), 12 *edges* (1-dimensional faces), and 6 *faces* (2-dimensional faces), so we say the $f$-vector is $(8, 12, 6)$.

Sample problems:

(1) Can $(9, 15, 8)$ be the $f$-vector of any polytope?
(2) What about $(50, 99, 50)$?

More generally, which $f$-vectors are possible and which are not? It turns out that it not too hard to completely answer this question in 3 dimensions, but in dimensions 4 and higher it is wide open.

*Prerequisites:* None, but linear algebra will be helpful.

*Homework:* Optional, but encouraged.

**Noncommutative ring theory.** (∗∗∗∗, Ari, week 1 of 1)

Rings are among the most ubiquitous algebraic structures in mathematics. Tragically, most children are taught at a young age that multiplication is commutative; that is, $x * y$ and $y * x$ represent the same quantity. However, many of the most important rings that you will find in nature are not commutative! We will study the basic properties of noncommutative rings and modules, culminating with the powerful Artin-Wedderburn theorem which classifies semisimple rings.

*Prerequisites:* As much group theory, linear algebra, and/or ring theory as possible.

*Homework:* Required.

*Related to:* Algebraic geometry (Weeks 1–2); Representation theory (Week 3); Category theory (Week 4)

## Colloquia (4-5 pm)

**John Conway.** NTBA.

## Visitor Bios

**John Conway.** (Princeton University—math)

One of the most creative thinkers of our time, John Conway is known for his ground-breaking contributions to such diverse fields as knot theory, geometry of high dimensions, group theory, transfinite arithmetic, and the theory of mathematical games. Outside the mathematical community, he is perhaps best known as the inventor of the "Game of Life".