Fundamentals Of Mathematical Foundations
Terminology, Tour, Induction, Set Theory, Lambda Calculus, Complexity, Monad, Number Theory

Like ships passing nearby on a foggy night, each oblivious to the presence of the other, up to recently most mathematicians and software engineers ignore the work of the other. This course is about removing the fog, and letting developers see what can be achieved in production environments using really good ideas from modern mathematics. More and more ideas from mathematics are beginning to seep into the world of programming. Many software innovations – from deep learning to 3D graphics to modern programming type systems - are based on modern mathematics. All developers have studied some mathematics at college, so this course builds on that. It should be considered a refresher, with an emphasis on practical application, to bring everyone up to speed with the basics and be ready to explore more advanced topics.

We pay particular attention to how mathematical ideas are presented. For example, saying "a monad is just a monoid in the category of endofunctors" utterly confuses developers, whereas we prefer "monads are programmable semicolons" (just used to insert custom code between each statement) is much clearer, yet equally accurate.

	Contents of One-Day Training Course	
	Review Of Fundamentals	**Set Theory**
	The language of mathematics (it is really not all Greek!)	In the past, set theory was considered the most suitable approach to the foundations of all of mathematics
Target Audience This course is aimed at modern developers who need a better grasp of how areas of mathematics can be practically applied to programming.	A mathematical object has certain properties and can be used in operations Mathematical structures are mathematical objects themselves that contains some arrangement of mathematical objects (often a set or similar, + something extra)	More recent approaches (e.g. type theory, category theory) are better However, set theory is still an important area and worth studying Deductive system based on first order logic Law of the excluded middle (LEM)
	Terminology Mathematical object (much broader use of 'object' term compared to programming) Symbols and (mathematical) variables Mathematical statement Proposition, expression, formula "a *proposition* is a statement susceptible to proof, whereas a *theorem* is such a statement that has been proven." (HoTT)	Axiom of choice, ... **Complexity Theory** Can a computational problem be solved? If so, how long will it take? Worst case scenario Big O notation
Prerequisites It is expected attendees will have completed some mathematics training as part of their college education.	**Branches of Mathematics ..** ..and their uses in software development Quantity/Arithmetic; Change/Calculus; Structure/Algebra; Space/Geometry Need to describe mathematical universes ..	**Monads** Isolating change in a non-changing world (think of carefully managed assembly lines - some steps make changes, others do not)
	Mathematical Whirlwind Tour Quick tour of all of mathematics So many aspects to it – where do we start? We like "The Map Of Mathematics" Which new/unfamiliar parts we should use	**Lambda Calculus** The three terms (and how they work) – variables, abstraction and application Untyped vs. simply-typed Extensions (exceptions, recursion, ..) Currying (higher order functions) How Lambda Calculus is used in modern programming languages/type systems
	Mathematical Induction Definition by induction, in steps Base step (e.g. 0) is the starting point Induction step: builds on base (n) Example of deductive reasoning	**Number Theory** Kinds of numbers (naturals, integers, reals, complex numbers) Broadening the scope (e.g. algebra) Specialist topics – e.g. Dedekind cut